Technical Report 371

J. E. Savage

The Computation Problem with Sequential Decoding

16 February 1965

Prepared under Electronic Systems Division Contract AF 19(628)-500 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lexington, Massachusetts

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-500.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

THE COMPUTATION PROBLEM WITH SEQUENTIAL DECODING

J. E. SAVAGE

Group 62

LINCOLN LABORATORY

TECHNICAL REPORT 371

RESEARCH LABORATORY OF ELECTRONICS

TECHNICAL REPORT 439

16 FEBRUARY 1965

LEXINGTON MASSACHUSETTS

THE COMPUTATION PROBLEM WITH SEQUENTIAL DECODING*

ABSTRACT

Fono Sequentiol Decading is o technique for communicoting ot o high infarmation rate ond with a high reliability over a large class of channels. However, equipment cast and variation in the time required to decade successive tronsmitted digits limit its use. This wark is cancerned with the latter limitation.

Others have shawn thot the overage pracessing time per decoded digit is small if the infarmation rate of the source is less thon a rote R_{camp}. This report studies the probobility distribution af the processing time rondom vorioble ond opplies the results ta the buffer overflow probability, i.e., the prabobility that the decoding deloy forces incoming data ta fill and overflow a finite buffer. It is shawn that the averflaw prababil ity is relatively insensitive to the buffer storage capacity and to the computational speed of the decader, but quite sensitive to informatian rate. In particular, halving the saurce rate mare than squores the overflow probability. These sensitivities are faund to be basic Sequential Decading and arise because the camputatian per decaded digit is large during an interval of high channel noise and graws exponentially with the length of such an interval.

A canjecture is presented cancerning the exact behaviar of the averflaw prababil ity with information rate. This conjecture agrees well with the (limited) experimentol evidence available.

Accepted for the Air Force
Stanley J. Wisniewski
Lt Calanel, USAF
Chief, Lincaln Laboratary Office

*This report is based on a thesis af the same title submitted ta the Department af Electrical Engineering at the Massachusetts Institute af Technalogy on 5 February 1965 in portial fulfillment of the requirements for the degree af Doctar af Philasophy.

TABLE OF CONTENTS

THE COMPUTATION PROBLEM WITH SEQUENTIAL DECODING

CHAPTER I
INTRODUCTION

A. BACKGROUND AND PREVIOUS WORK

The branch of statistical communication theory known as coding theory has received much attention since the results of C. E. Shannon[1] in 1948. Many investigations were and are attracted to coding theory because of the potential for ultrareliable communication suggested by Shannon's Noisy Coding Theorem. Loosely stated, this theorem says that data can be encoded for transmission over a noisy channel in such a way that the probability of a decoding error is arbitrarily small, provided that the information rate of the source is less than a rate called channel capacity; the converse to the Noisy Coding Theorem essentially says that channel capacity is the largest rate at which the probability of error can be made arbitrarily small.

The implications of the Coding Theorem are obviously stimulating. The fact that codes exist for noisy channels which achieve small error probabilities while operating at a fixed information rate is quite surprising. A priori, one would have expected that reliability could be achieved only by repeating the transmitted message, that is, that reliability is obtainable only at the expense of less information per unit time, i.e, a reduction in rate.

Although the Coding Theorem indicates the potential for ultrareliable communication, it has been found that this ultrareliability costs either a great deal in equipment or in decoding delay. Both costs are exorbitant if the decoder operates so as to strictly minimize error probability. Practical considerations force one to consider less than optimum codes and decoders (in a probability of error sense). A number of such codes and decoders have been invented. Included among these various coding techniques are Massey's Threshold Decoding,[2] Gallager's Low Density Parity Check Codes,[3] Bose-Chaudhuri Codes with the Peterson Decoding Procedure,[4] Iterative Decoding,[5, 6] and Sequential Decoding[7, 8] as first presented by J. M. Wozencraft and later modified by R. M. Fano. Each of these procedures and others[9, 10] not mentioned find application depending upon the performance requirements which are set and the economics of the application. Sequential Decoders score reasonably well in both the performance and economic categories. We shall concentrate on Sequential Decoding, and in particular on the Fano Sequential Decoding Algorithm, in this report.

B. FORMULATION OF PROBLEM

In many ways, the Fano algorithm is an attractive decoding procedure. It applies to a large variety of channels in contrast with the algebraic codes such as Bose-Chaudhuri codes which are best adapted to symmetric channels with an equal number of inputs and outputs (which is a power of a prime[4]). The Fano algorithm is also recommended by the fact that it will operate with high reliability at a substantial fraction of channel capacity. Thus, it is ideally suited for systems handling high-quality, high-volume traffic.

The Fano algorithm, however, possesses at least two disadvantages. The first is that the necessary encoding and decoding equipment is expensive. The second and most damaging disadvantage of the Fano algorithm is that the time required to process the incoming data is variable and assumes very large values during intervals of high channel noise. The variability of the processing time requires that incoming data be buffered. The fact that this processing time assumes large values implies that occasionally and eventually a finite buffer will fill and overflow. After overflow, it is found that the decoder often performs erroneously. Such an event is catastrophic unless moderated with periodic resynchronization, the use of a feedback channel, or some other means.

Not only is overflow serious, but it occurs much more frequently than do undetected decoding errors (i.e., errors without overflow). Thus, it is the controlling event in the design of the decoder. Although the overflow event is serious, the decoder can be so designed and the information rate be so restricted that overflows are very infrequent. It is, therefore, a problem which can be resolved.

Our concern in this report is to obtain some understanding of the sensitivity of the overflow probability to the following: the buffer capacity, the machine speed and the information (or signaling) rate. This is a difficult analytical problem. As a result, we have been forced to analyze the machine performance and to determine the various sensitivities indirectly. Our approach to the overflow question has been to consider a random variable of computation (called "static" computation) which is related to the computation performed by the machine as it decodes. We have shown that the cumulative probability distribution function P_R $[C \geqslant L]$ of the random variable of "static" computation C is an algebraic function of the distribution parameter L, that is, it behaves as $L^{-\alpha}$, $\alpha > 0$, for large L. From this behavior and a study of the exponent α, we have found through a heuristic argument that the probability of buffer overflow is relatively insensitive to a change in machine speed or to the size of the buffer but that it is quite sensitive to information rate, being more than squared by a halving of rate.

The deductions on the sensitivities of the overflow probability indicate that practical limits on the size and speed of a decoder are set primarily by the overflow probability and that the machine performance is really only sensitive to information rate. This sensitivity is due to the fact that P_R $[C \geqslant L]$ behaves as $L^{-\alpha}$ for large L. We shall find that P_R $[C \geqslant L]$ behaves as $L^{-\alpha}$ for large L because for every transmitted codeword there exists an interval of high channel noise such that "static" computation is large and growing exponentially with the length of the interval of high channel noise. The probability of such a noisy interval decreases exponentially with the length of the interval. It is the balance between the two exponentials which forces the algebraic nature of the distribution of "static" computation, P_R $[C \geqslant L]$. Since this same balance is fundamental to the entire concept of Sequential Decoding, it does not appear that the buffer overflow problem can be avoided unless some major modification of the decoding procedure can be devised.

These results and arguments are explained in detail in the following chapters.

Chapter II focuses on the Fano Sequential Decoding Algorithm. The algorithm is defined, motivated and discussed. Many of its properties are clearly outlined. The buffer overflow problem is discussed and the random variable of "static" computation is defined.

Chapter III is prefaced with a discussion of the connection between an exponential growth in computation with the length of an interval of high channel noisc and the algebraic nature of the distribution of "static" computation. The main purpose of the chapter is to underbound the distribution of "static" computation. A general underbound is found which applies to all codes on the "completely connected" discrete memoryless channel (DMC). A lower bound is also found for the (small) subset of codes which have fixed composition, again for the "complctely connected" DMC. Both bounds to $P_R\ [C \geqslant L]$ are algcbraic in L.

Chapter IV concentratcs on obtaining an upper bound to the distribution of "static" computation, $P_R\ [C \geqslant L]$. Since there are "poor" codes, codes for which $P_R\ [C \geqslant L]$ is large so that largc computation occurs with high probability, we must establish that codes exist with a $P_R\ [C \geqslant L]$ which decreases as an algebraic function in L. (It cannot decrease any faster because of the lowcr bound result.) We show that such codes exist by averaging $P_R\ [C \geqslant L]$ over the enscmble of all tree codes. This average is algebraic in L so that many codes exist with an algebraic distribution function.

Chapter V interprets the upper and lower bounds to $P_R\ [C \geqslant L]$, describes an experiment performed at Lincoln Laboratory and compares the results of this experimcnt to the tail behavior of $P_R\ [C \geqslant L]$, i.e., its behavior for large L. The comparison leads to a conjecture on the truc tail behavior of $P_R\ [C \geqslant L]$. It is shown that this conjecture has a very close connection to some fundamental results in information theory which are expressed in the Coding Theorem. Finally, a heuristic connection between the distribution of "static" computation and the overflow probability is established and the sensitivity of the overflow probability to machine specd, buffer size and information rate is brought out. Some problems deserving further research are also suggested.

CHAPTER II
DESCRIPTION OF FANO SEQUENTIAL DECODING ALGORITHM

This chapter briefly discusses the encoding problem and introduces the Fano Sequential Decoding Algorithm.[8] The dynamics of the algorithm are described and a definition of computation is presented. This chapter serves as preparation for the following analytical chapters.

A. TREE CODES

Let us assume that the output of a source with a b-letter alphabet is encoded for transmission on a discrete memoryless channel (DMC). (The DMC is characterized by the set of transition probabilities $\{p(y_j/x_k)\}$ where $\{x_k\}$, $1 \leqslant k \leqslant K$ is the channel input alphabet and $\{y_j\}$, $1 \leqslant j \leqslant J$ is the channel output alphabet.) Consider encoding the source by mapping a sequence of source digits into a sequence of channel digits. The channel digits are selected from an array that topologically resembles a tree and will henceforth be called a tree (see Fig. 1).

For the moment, consider mapping a finite sequence $\overline{\beta}_n = (\beta_1, \beta_2, \ldots, \beta_n)$ of n digits drawn from the source alphabet onto a finite channel sequence $\overline{u}_n = (\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_n)$, where $\underline{u}_q = (u_{q1}, \ldots, u_{q\eta}, \ldots, u_{q\ell})$ is the subsequence of ℓ digits (or a tree branch) drawn from the channel input alphabet. At the qth node of the tree, β_q directs a path along the bottom branch if $\beta_q = a_1$,

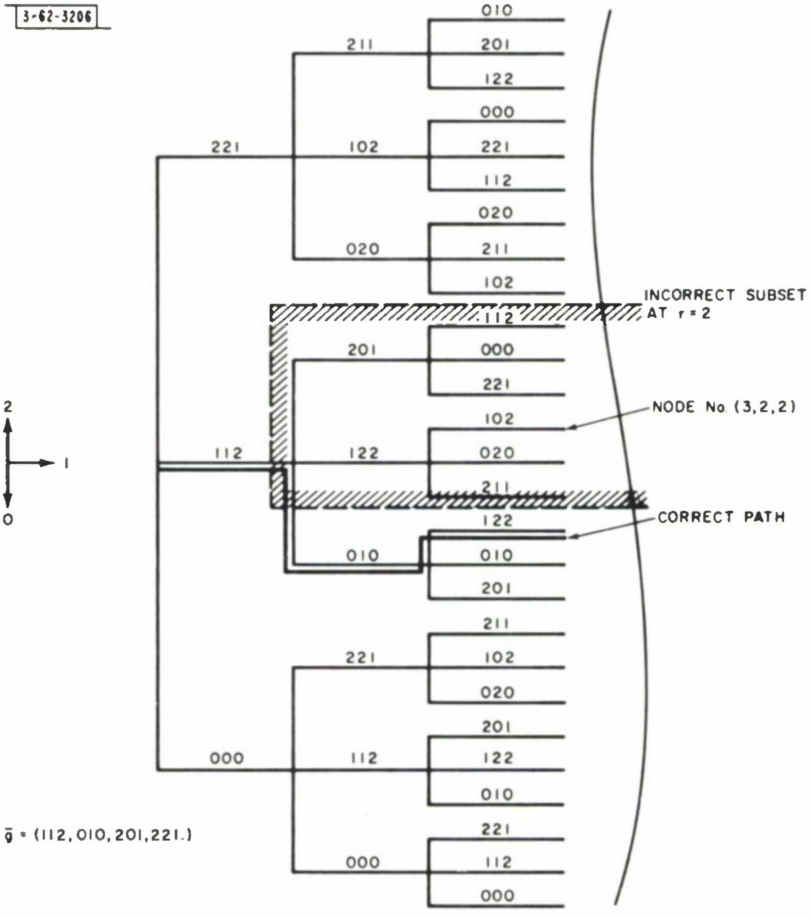


Fig. 1. Convolutional tree code.

along the second branch from the bottom if $\beta_q = a_2$, and along the top branch if $\beta_q = a_b$. (A path is a contiguous sequence of branches.) For example, the channel input sequence $\bar{u}_3 = (112, 010, 122)$ corresponds to the source sequence $\underline{\alpha}_3 = (1, 0, 2)$ in Fig. 1 when the source and channel input alphabets are both $\{0, 1, 2\}$.

The extended source sequence $\bar{\beta}(=\bar{\beta}_\infty)$ specifies an infinite path $\bar{u}(=\bar{u}_\infty)$ through the tree. The path $\bar{u}$ will be called the correct path. For each node of the correct path, say the q^{th}, $q = 0, 1, 2, \ldots,$ where the 0^{th} node is the origin, we define an "incorrect subset." The incorrect subset at the q^{th} node consists of (1) the q^{th} node itself and (2) all nodes (of depth greater than q) diverging from the q^{th} node, which are not part of the correct path. For example, see Fig. 1 where the incorrect subset at the 2^{nd} node of the correct path is shown.

We shall find it useful to classify nodes in each incorrect subset. Consider the q^{th} such subset. Consider a node "at penetration s" in this subset (such a node is the terminus of a path of q + s branches). There are a number of nodes at this penetration s. Let the node in question be m^{th} from the bottom of this set of nodes. Then, it is uniquely identified by the triplet (m, s, q). This triplet indicates that the particular node is m^{th} in rank among nodes at penetration s in the q^{th} incorrect subset (see Fig. 1). The q^{th} node of the correct path (or the reference node) is identified by the triplet (1, 0, q). (By convention, this single node is said to be at penetration zero in the q^{th} incorrect subset.) Denote by M(s) the number of nodes at penetration s in the q^{th} incorrect subset. Then,

$$\begin{aligned} M(0) &= 1 \\ M(1) &= (b-1) \\ M(2) &= (b-1)\,b \\ &\ \ \vdots \\ M(s) &= (b-1)\,b^{s-1} \quad \text{for} \quad s \geqslant 1 \quad . \end{aligned} \tag{1}$$

There are M(s) paths at penetration s in the q^{th} incorrect subset, and each of these paths contains q + s branches.

Given that $\bar{u}_n = (\underline{u}_1, \ldots, \underline{u}_n)$ is transmitted, let $\bar{v}_n = (\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_n)$ be the received sequence, where $\underline{v}_q = (v_{q1}, \ldots, v_{q\eta}, \ldots, v_{q\ell})$ is the q^{th} subsequence of ℓ channel output digits. The probability that $\bar{v}_n$ is received when $\bar{u}_n$ is transmitted is computed from the transition probabilities of the DMC as follows:

$$P_R\,[\bar{v}_n/\bar{u}_n] = \prod_{q=1}^{n} P_R\,[\underline{v}_q/\underline{u}_q] = \prod_{q=1}^{n}\prod_{h=1}^{\ell} p\,[v_{qh}/u_{qh}] \tag{2}$$

where $p\,[v_{qh}/u_{qh}] = p\,[y_j/x_k]$ when $v_{qh} = y_j$ and $u_{qh} = x_k$.

The data (or signaling) rate (in bits per channel use) is defined as

$$R = \frac{\log_2 b}{\ell} \quad . \tag{3}$$

If the successive source digits are equally likely and statistically independent, then R is also the source entropy (or information rate) per transmitted digit. We shall assume that successive source digits meet these conditions.

B. CONVOLUTIONAL CODES

Although we shall later assume for analytical convenience that data are encoded with an arbitrary tree code, we present convolutional codes here to show that tree codes may be realized with a minimum of equipment.

Define a basic sequence $\bar{g} = (\underline{g}_1, \underline{g}_2, \dots, \underline{g}_S, \underline{0}, \underline{0}, \dots)$, called the code generator, where $\underline{g}_r = (g_{r1}, \dots, g_{r\ell})$ is the r^{th} subsequence of ℓ digits, and S is called the code constraint length. We also define translates of $\bar{g}$ by

$$\bar{g}_n = (\overbrace{\underline{0}, \underline{0}, \dots, \underline{0}}^{n}, \underline{g}_1, \dots, \underline{g}_S, \underline{0}, \dots)$$

where $\underline{0}$ indicates a subsequence of ℓ zeros. Assume that the letters in the generator $\bar{g}$ and the letters of the source alphabet coincide and consist of the set of integers $\{0, 1, \dots, b-1\}$ b a prime. Then, the source sequence $\bar{\beta} = (\beta_1, \beta_2, \dots)$ generates the channel sequence $\bar{u} = (\underline{u}_1, \underline{u}_2, \dots)$ by

$$\bar{u} = \sum \beta_n \bar{g}_n \quad . \tag{4}$$

Multiplication and vector addition are taken modulo b. Following this rule the tree, partially shown in Fig. 1, may be constructed from the code generator $\bar{g} = (112, 010, 201, 221, 000, \dots)$. In particular, the source sequence $\bar{\beta} = (1, 0, 2, \dots)$ generates the channel sequence $\bar{u} = (112, 010, 122, \dots)$.

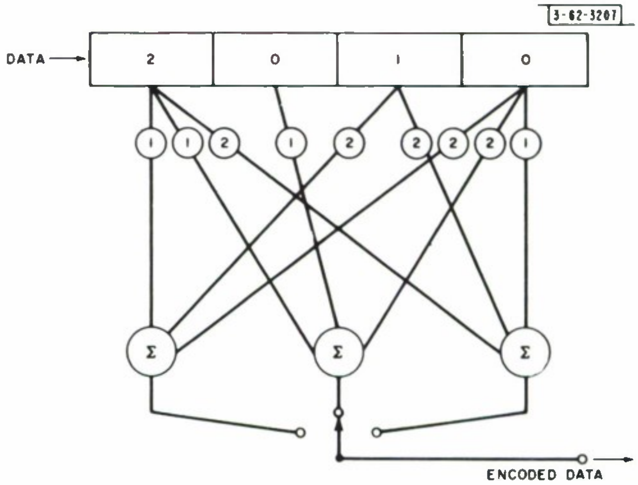


Fig. 2. Convolutionol encoder.

This code can be realized (see Fig. 2) with a standard shift register of S stages (the code constraint length), multipliers† and adders (modulo b). Clearly, the size of the convolutional encoder does not increase faster than linearly in the code constraint length. Others have shown that the probability of a decoding error with Sequential Decoding on convolutional codes decreases exponentially in the code constraint length (for almost all codes). In a probability of error sense, convolutional codes are near optimum.

† The circles in Fig. 2 indicote multiplication by the enclosed numbers.

This example has assumed that the source alphabet and channel alphabet are identical. Neither this restriction nor the restriction that the alphabets contain the same number of elements is needed (see Ref. 11). In addition, the constraint that b be prime is not essential. For example, b may be a power of a prime and the components of $\overline{\beta}$ and $\overline{g}$ may be chosen as elements of a general Galois field, addition and multiplication taken in this field.[4]

C. FANO ALGORITHM

In preparation for a discussion of the Fano search procedure, we introduce and motivate the "metric" with which the procedure operates.

1. Metric

Assume that a source generates a sequence of outputs $\overline{\beta}$. This sequence directs a path $\overline{u}$ through a tree code. The branches of this path are transmitted over a discrete memoryless channel. A sequence of branches $\overline{v}$ is received at the channel output. The Fano decoder is a device that operates on this sequence and produces a replica of the transmitted sequence, unless decoding errors occur.

The Fano decoder (or algorithm) is a rule for searching efficiently through the paths in the tree code in an attempt to find a "best fit" with the received sequence $\overline{v}$. To determine a "best fit," values are assigned to nodes in the tree. The value of a node is said to be the value of the metric between the path terminating on this node and the corresponding received sequence. As the decoder searches nodes, values of the metric are compared to the criteria of Fig. 3. The criteria $T_i = i\,t_o$ are straight lines of zero slope separated by an amount t_o.

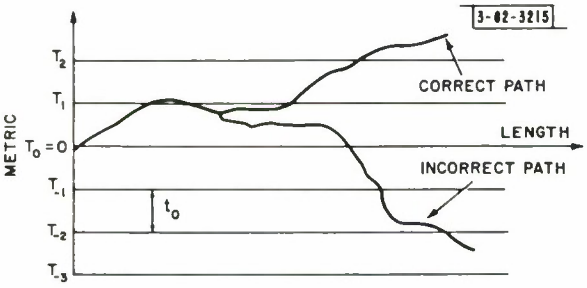


Fig. 3. Criteria and typical paths.

Let us be precise about the definition of metric. We define a "branch metric" and associate a value of this branch metric with each branch of the tree.† Let $\underline{u}_o = (u_{o1}, u_{o2}, \ldots, u_{o\ell})$ be a tree branch and let $\underline{v}_o = (v_{o1}, v_{o2}, \ldots, v_{o\ell})$ be the corresponding received branch. The branch metric between $\underline{u}_o$ and $\underline{v}_o$, $\underline{d}(\underline{u}_o, \underline{v}_o)$, is defined as

$$\underline{d}(\underline{u}_o, \underline{v}_o) \triangleq \sum_{h=1}^{\ell} [I(u_{oh}, v_{oh}) - R] \qquad (5)$$

† This is not a metric in the mathematical sense because $\underline{d}(\underline{u}_a, \underline{v}_a)$ may be negative.

where†

$$I(u_{oh}, v_{oh}) \triangleq \log_2 \frac{p\,[v_{oh}/u_{oh}]}{f(v_{oh})} \quad . \tag{6}$$

Here, $p\,[v_{oh}/u_{oh}] = p\,[x_j/x_k]$ when $v_{oh} = y_j$, and $u_{oh} = x_k$. We let $f(v_{oh})$ be a probability-like function. It may be interpreted as the probability of channel output symbol v_{oh} when the channel inputs are assigned probabilities $\{p_k\}$, $1 \leqslant k \leqslant K$. The function $f(v_{oh})$ and the probability assignment $\{p_k\}$ will appear during the "random code" argument of Chapter IV and an interpretation will be attached to $f(y_j)$ and $\{p_k\}$.

The "path metric," d(m, s, q), on the path containing q + s branches and terminated by node (m, s, q), is defined as the sum of thc branch metric on each of the q + s branches. The value of this path metric is associated with node (m, s, q). When we plot d(m, s, q) for paths in the tree, we indicate the values of the path metric with nodes. The nodes in this plot have a one-to-one correspondence to nodes in the tree and will be indexed with the same triplet (m, s, q).

This definition of path metric is justified by two facts – it leads to a workable decoder and this decoder can be studied analytically. The definition is recommended by the fact that a large value of the path metric indicates that the path in question is very probable a posteriori (see below) which is equivalent to saying that with high probability this path is the transmitted path. We now show that the value of the metric is monotone increasing in the a posteriori probability of a path.

Let $\bar{u}_n$, $n = q + s$, represent the tree path (m, s, q) and let $\bar{v}_n$ bc the corresponding received sequence. Then, the value of the metric on $\bar{u}_n$ is

$$\begin{aligned} d(m, s, q) &\triangleq \sum_{r=1}^{n} \sum_{h=1}^{\ell} [I(u_{rh}, v_{rh}) - R] \\ &= \log_2 \frac{P_R\,[\bar{v}_n/\bar{u}_n]}{f(\bar{v}_n)} - n\,R \end{aligned} \tag{7}$$

where u_{rh}, v_{rh} are the h^{th} digits on the r^{th} branch of $\bar{u}_n$, $\bar{v}_n$, respectively, and

$$f(\bar{v}_n) \triangleq \prod_{r=1}^{n} \prod_{h=1}^{\ell} f(v_{rh}) \quad . \tag{8}$$

In obtaining Eq. (7), we have used Eqs. (4) and (5), together with the definition of $P_R\,[\bar{v}_n/\bar{u}_n]$ of Eq. (2). Now, $P_R\,[\bar{v}_n/\bar{u}_n]$, the conditional probability that $\bar{v}_n$ is received when $\bar{u}_n$ is transmitted, is proportional to $P_R\,[\bar{u}_n/\bar{v}_n]$, the a posteriori probability of $\bar{u}_n$, since (from Baye's Rule)

$$P_R\,[\bar{u}_n/\bar{v}_n] = P_R\,[\bar{v}_n/\bar{u}_n]\,\frac{P_R\,[\bar{u}_n]}{P_R\,[\bar{v}_n]} \tag{9}$$

and $P_R\,[\bar{u}_n]$, the a priori probability of $\bar{u}_n$, is constant under variation of $\bar{u}_n$. (We have assumed that successive source digits are statistically independent and identically distributed.) Thus, we have established for the given source that the path of n branches with the largest valuc of the metric is that path of n branches which is a posteriori most probable.

† If output y occurs with probability f(y) then I(x,y) is the "mutual information" between x and y.

We have attached a value of the branch metric to each of the b branches stemming from a node. We observe by analogy with the argument above, that of these branches, that branch with the largest value of the branch metric is the a posteriori most probable branch at that node. Then, we order branches at a node according to their value of the branch metric and say that they are "most probable," "second most probable," etc.

We consider next the motivation for and definition of the Fano algorithm.

2. Search Procedure

Sequential Decoding procedures in general, and the Fano algorithm in particular, are motivated by the following consideration: For a properly chosen code and for signaling rates which are suitably restricted, the a posteriori probability of the correct path and the value of the path metric on it will typically increase (see Fig. 3). On the contrary, any incorrect path branching from the correct path will typically decrease in path metric (see Fig. 3). Thus, a separation will typically occur between the correct path and some incorrect path. Using a set of thresholds, a decoder can eliminate from consideration a large number of improbable, hence, incorrect paths. As long as the channel "noise" is not too severe, the separation between the correct and incorrect paths will become increasingly evident. A period of high channel noise, however, may force a large amount of searching and even cause decoding errors. We shall consider these two points later.

The set of rules for searching tree paths which we shall consider here is known as the Fano Sequential Decoding Algorithm. A logical flow chart of this procedure[†] is given in Fig. 4. To

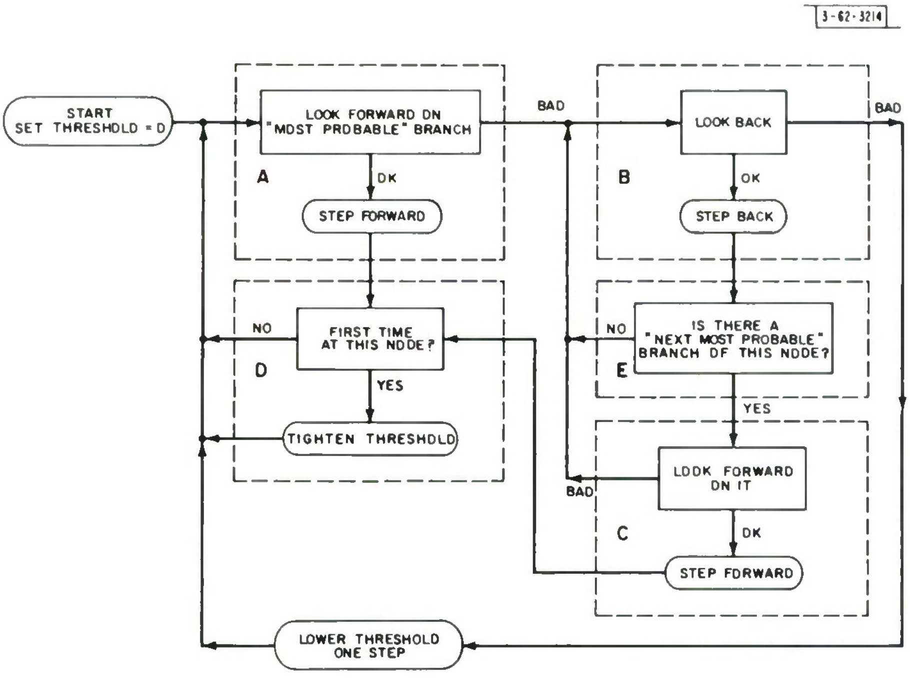


Fig. 4. Flow chart of Fano algorithm.

[†] See Ref. 8 far the flaw chart of the computer program which realizes the chart of Fig. 4. The bookkeeping required by D of Fig. 4 is accamplished with a small number af instructions in the camputer program. This chart is based an a flaw chart suggested by Professor I. M. Jacabs.

describe the operation of this algorithm we introduce the notions of forward mode and search mode operations. The machine operates in the forward mode when it is searching for the first time a path whose metric is nondecreasing. (We shall be more precise about this point later.) Roughly speaking, the machine operates in the search mode when it is looking for a path which has a continuously growing metric.

Let us now be specific. Suppose that the decoder is following a path which is growing in metric and that this path is being followed for the first time so that the machine is operating in the forward mode. Then, at each node of this path the decoder raises a threshold, called the running threshold T in units of t_o until it lies just below the value of the path metric at each node. In Fig. 4 this operation is performed by D. After the threshold is tightened at a node, the decoder looks forward along the "most probable" branch (that one which has the largest value of the branch metric). If the path metric on the extended path remains above the existing value of the running threshold T, and if the extended path is examined for the first time, forward mode operation continues. If the extended path falls below T, as in Fig. 5, search mode operation begins. Operation B of Fig. 4 is then performed.

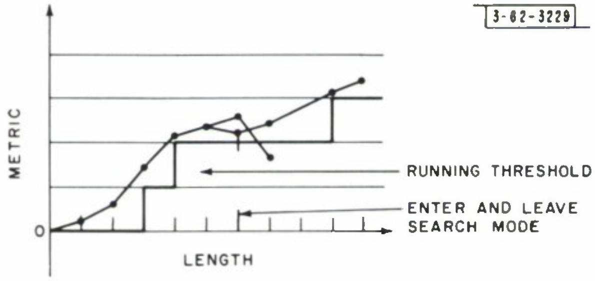


Fig. 5. Typical machine search.

When the machine enters B it is looking for a path which will remain above T. Hence, it looks back to the preceding node to determine whether it remains above T. If so, (OK) perhaps the "next most probable" branch extending forward from the original node will remain above T. At E, the machine determines whether a "next most probable" node exists, and if not, it looks back again with the same intention, that is, of finding an extendable path. If in looking forward in C the machine finds that the extended path remains above T, it steps forward tightening the running threshold if this node is visited for the first time. (This threshold is tightened and the machine enters or remains in the forward mode only when a node is examined for the first time. Otherwise, looping would occur.) If the forward look in C is successful, the machine steps forward and continues to look forward, as indicated by Fig. 5. If the forward look in C is unsuccessful, the machine again looks back in search of a node from which an extendable path may be found (i.e., a sequence of nodes which remains above T). If an extendable path cannot be found, that is, if every sequence remaining above T and connected to the node at which searching begins eventually crosses T, then the running threshold T must be reduced. After the threshold is reduced, the decoder looks forward along "most probable" branches until it reaches the node at which it entered the search mode. The branch on which the decoder looks forward is a new branch, so that the threshold may be increased if this extended path lies above T (see Fig. 6).

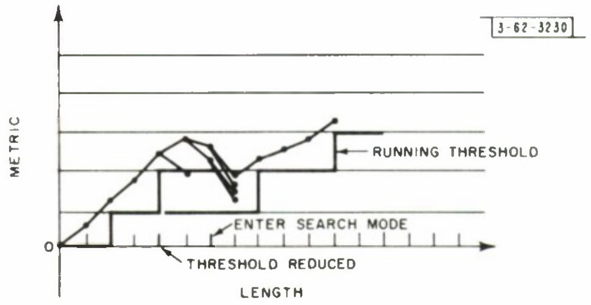


Fig. 6. Threshold reduction, b = 2.

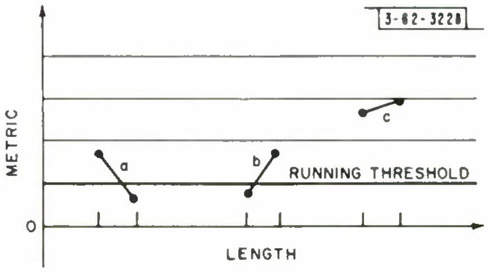


Fig. 7. Branch examination with a threshold.

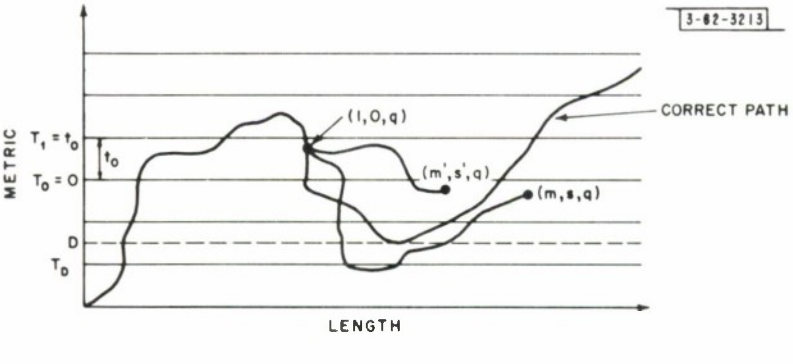


Fig. 8. Minimum threshold T_D.

The machine operation may be summarized as follows: The decoder operates in the forward mode, extending along "most probable" branches and increasing the running threshold as it progresses, until an extension fails the running threshold T. At this point, search mode operation begins and the decoder looks for a sequence of nodes which remains above T. If each sequence of nodes connected to the node at which search mode operation began is such that it crosses T before forward mode operation resumes, then T is reduced. As soon as the decoder finds a new path remaining above the existing value of T, forward mode operation begins and T may be increased.

D. COMPUTATION

We now establish that the decoder does not look forward or back on any given branch more than once with each value of the running threshold. There are three situations which need to be considered. There is a node at each end of the given branch. We need to consider the case where both nodes lie above a given threshold, and where either the preceding or following node lies below the given threshold. If both nodes fall below some threshold, the branch considered will not be examined with this threshold.

If the node preceding the branch in question lies above the given threshold, while the following node lies below this threshold (see a of Fig. 7), then the decoder may look forward on this branch, but it cannot look back because it would have to step forward to do so. But from A of Fig. 4, it cannot step forward while this threshold is in effect. Next consider the situation of b in Fig. 7. The decoder can look back on the given branch, but it cannot look forward because it would have to step back to do so, which is prevented by the restriction OK in B of Fig. 4. The third situation to be considered is that of c in Fig. 7. Both nodes terminating the branch in question lie above the given threshold. With this threshold the decoder may look forward and then step forward (A of Fig. 4) from the preceding to the following node. The decoder may then search forward and later return to the second node with this same threshold. We now show that the decoder cannot return to the first node and then retrace this branch. We observe from B, E, and C of Fig. 4 that this branch with the given threshold cannot be retraced because the decoder can extend only along either the "next most probable" branch at the first node, or along the "next most probable" branch at an earlier node. The decoder can only retrace the original branch by exiting from B on BAD (Fig. 4) and lowering the threshold. Thus, with any given threshold any particular branch cannot be examined in the forward and reverse directions more than once.

Now let us consider the lowest threshold which is used by the decoder. Consider paths branching from the q^{th} node of the correct path and terminating on nodes labeled (m, s, q), $1 \leqslant m \leqslant M(s)$, $0 \leqslant s < \infty$. Let D be the correct path minimum at or following the q^{th} node and let T_D be the threshold just below D† (see Fig. 8). Assume that the received path is decoded correctly, that is, that decoding errors are not made. Then paths which cross T_D will not be examined beyond the point at which they cross T_D. This is true since threshold $T_D - t_o$ is used only if all paths fall below T_D; but by definition the correct path remains above T_D. This implies that the decoder will not step forward to a node which lies below T_D nor to any node connected to and following such a node (see (m, s, q) of Fig. 8).

† Since the decoder operation depends only on incremental values of the metric, we may assume that the q^{th} correct node lies between T_a and T_1, and measure D and T_D from $T_o = 0$.

We may also deduce that if $D < 0$ and all nodes connecting any node such as (m', s', q) in Fig. 8 to $(1, 0, q)$ [including (m', s', q)] be above $T_D + t_o$, then the decoder must look forward from (m', s', q) before the threshold is reduced to T_D. (The constraint $D < 0$ is necessary because if $D \geqslant 0$ the machine may never be forced back to $(1, 0, q)$ so that forward or backward looks from (m, s, q) may never occur.)

The two central results of the last three paragraphs may be summarized as follows:

(1) Consider a node (m, s, q) branching from the q^{th} node of the correct path. Let D be the correct path minimum on or following the q^{th} node. Let T_D be the threshold just below D. Assume that node (m, s, q) lies between thresholds T_{n+1} and T_n where $T_n \geqslant T_D$ as in Fig. 9. Let N_k be the number of forward or backward looks from this node with threshold T_k. Then, for each threshold $T_k \geqslant T_D$ and $T_k \leqslant T_n$, $T_n \geqslant T_{n-1} \geqslant \ldots \geqslant T_k \geqslant \ldots \geqslant T_D$, we have

$$0 \leqslant N_k \leqslant b + 1$$

N_k is zero for any other threshold. The lower limit represents a situation of the type represented by (m, s, q) in Fig. 8; in this case, the machine does not look forward or backward from the node in question.

The conditions under which $N_k = 0$ and the bounds on N_k in Eq. (10) are central to the arguments of Chapter IV, which is concerned with overbounding the statistics of the decoder behavior.

(2) Consider a node such as (m', s', q) of Fig. 8. This node remains above $T_D + t_o$ and is connected to $(1, 0, q)$ through a set of nodes all of which lie above $T_D + t_o$. If $D < 0$, the decoder must look forward at least once from this node before the threshold T is reduced to T_D (to which it must be reduced, since the decoded path is the correct path and this path lies below $T_D + t_o$ at some point).

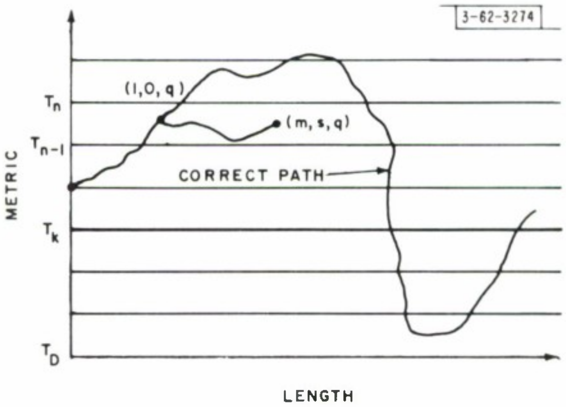


Fig. 9. Typical path trajectories.

The conditions under which the decoder must look forward at least once from node (m', s', q) are central to the arguments of Chapter III, which is concerned with underbounding the statistics of the behavior of the decoder.

We shall call the number of forward and backward looks at a node the "computation" at this node. These looks are the operations which require machine time. In the remainder of this report, we use this definition of computation to investigate the computational demands of the decoder.

E. BUFFER AND DYNAMICS OF DECODER

In the previous section, we assumed implicitly that the decoder is capable of searching back indefinitely into the tree in the process of decoding. Although this assumption will be needed for later analysis, it is not consistent with a physical machine. To search back indefinitely requires that all received branches be stored in the decoder. Practical limitations on the cost and size of the decoder force one to consider buffers for storage which are of finite size. We shall consider now a particular buffer realization and discuss the dynamics of the decoder operation.[12]

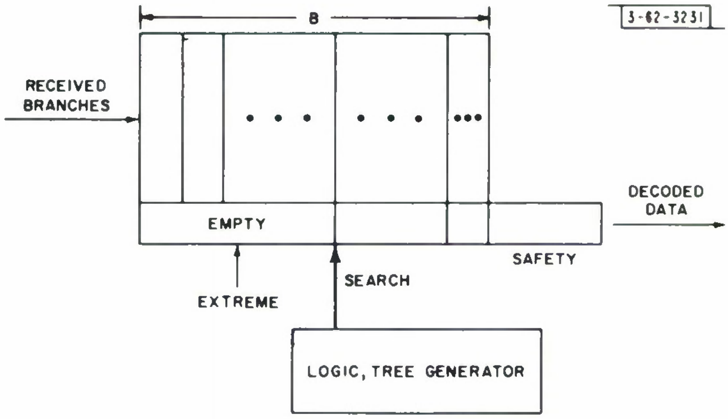


Fig. 10. Buffer.

Assume that the decoder operates with the buffer of Fig. 10. Received branches are inserted at the left end of the buffer and progress through the buffer at the rate at which they arrive. The buffer stores B branches. Below each branch there is space to register an element of the source alphabet. As the decoder operates, it inserts into these places tentative source digit decisions. Insertions are made at the position of the "search" pointer. When these tentative decisions reach the left-hand side of the safety zone they are considered to be final. When they reach the right-hand side of the safety zone they are considered to have been decoded. If a digit released from the right end of the safety zone disagrees with the corresponding source output digit, a decoding error is said to have occurred.

The "search" pointer indicates the received branch at which the decoder is looking. The "extreme" pointer indicates the most recently received branch that has been examined. As the machine operates the two pointers may advance together toward the left-hand side of the buffer until a search is necessary. At that time the search pointer and the extreme pointer will drift back, the search pointer moving away from the extreme pointer. (When the extreme pointer is not moving forward, it drifts back because branches are arriving at a constant rate.) As the seareh pointer moves back it erases previous tentative deeisions, and in moving forward it introduees new tentative decisions. Digits in the safety zone cannot be changed.

It has been found from simulation[13] that under normal operating conditions the two pointers usually hover near the left-hand side of the buffer. Occasionally, however, they will drift back a substantial distance. During this drift, the two pointers usually are separated by a small fraction of the distance they have drifted from the buffer end. (This behavior is rationalized by the observation that the number of machine computations tends to grow exponentially with the depth of the search from the extreme point.)

Occasionally, the search pointer reaehes the far end of the buffer. Then, the decoder is likely to release an incorreet digit into the safety zone; thereafter, the decoder tries to advance on an incorrect tree path. Since this is difficult, the machine must do a large amount of computation. The search pointer then hovers near the far end and additional erroneous source digits are released into the safety zone. Thus, if the search pointer is forced to the far end of the buffer, it will tend to remain at this end and to decode in error. We call this event buffer overflow. This report is motivated by a concern for this event.

Although, deeoding errors may oecur without causing a large machine computation, it is noted from simulation[13] (and may be rationalized heuristically) that for safety zones of moderate size, decoding errors are almost always preceded by overflow. The heuristic argument states, in effect, that the noise sequences, which are responsible for errors in the absence of overflow, occur with vanishingly small probability, especially for safety zones of large capacity.

Since buffer overflow can be detected, the decoder ean discard the unreliable digits in the safety zone. Thus, the probability that an erroneous digit is released to the user before the buffer overflows can be made very small, much smaller than the probability of overflow. This observation is equivalent to the statement that the probability of a machine failure, where failure means overflow or error is dominated by the probability of buffer overflow. Represent this probability by $P_{BF}(N)$. We define $P_{BF}(N)$ as the probability that the first buffer overflow occurs on or before the time at which the N^{th} source decision enters the safety zone.

We shall be concerned in this report with the sensitivity of P_{BF} to buffer size B, to the speed of the decoder and to the data rate R. We shall find that P_{BF} is relatively insensitive to buffer size and machine speed, but quite sensitive to data rate. We shall establish the mechanism which is responsible for the particular sensitivities of P_{BF}. Throughout, we assume that the decoder is working with a fixed channel.

A preliminary statement can be made here concerning the largest signaling rate R at which P_{BF} is "small" or at which the decoder will function well. Others[7,8,13,14] have shown, through analysis and simulation, that the largest rate at which the average computation per decoded digit is small is a rate called R_{comp}. Since large average computation implies frequent buffer overflows, R_{comp} is an upper limit on the rate at which the machine will function properly. R_{comp} is strictly less than channel capacity, except for pathological channels, and is a large fraction of channel capacity for many but not all channels.

F. "STATIC" COMPUTATION

Unfortunately, the statistics of the dynamical computation performed by the Fano decoder as it operates in time are too difficult to study directly through analysis, Consequently, we are led to consider a kind of computation called "static" computation which is at once analytically tractable and closely connected to the real machine computation. Through an investigation of "static" computation, we shall be able to make strong qualitative statements about the sensitivities of P_{BF}.

A restriction to the study of "static" computation has been found necessary without exception by all others who have investigated the Fano algorithm.[14-16] By "static" computation we mean a computation which is eventually performed by the decoder, if no digits are decoded in error and if the buffer is infinite. Thus, the assumptions are that the decoder has a buffer of infinite capacity, that it has operated for an indefinite length of time, and that it has decoded correctly.

Let (m, s, q) be a node of the q^{th} incorrect subset where $1 \leqslant m \leqslant M(s)$, $0 \leqslant s < \infty$, and M(s) is given by

$$M(0) = 1$$

$$M(s) = (b-1)\, b^{s-1} \quad , \quad \text{for } s \geqslant 1 \qquad \text{[Eq. (1)]}$$

We define "static" computation associated with the q^{th} correct node as the number of computations made on each node (m, s, q) of the q^{th} incorrect subset.

The connection between "static" computation and the probability of overflow will be made later.

CHAPTER III
LOWER BOUND TO DISTRIBUTION OF COMPUTATION

In this chapter, we underbound the cumulative probability distribution of the random variable of "static" computation C, namely, $P_R\,[C \geqslant L]$. This underbound applies to discrete, memoryless channels (DMC) which are completely connected (all channel transition probabilities are strictly positive). We show that this lower bound is an algebraic function of the distribution parameter L for large L; that is, $P_R\,[C \geqslant L] \geqslant (A/L^{\alpha})$ for all L greater than some constant L_o, where A, $\alpha > 0$.

The lower bound derivation is preceded by a discussion of the condition on the random variable of "static" computation which is responsible for its having an algebraic distribution function. Roughly speaking, this condition states that the distribution is algebraic if "static" computation is large during an interval of high channel noise and grows exponentially with the length of such an interval. This important result is responsible for the particular sensitivities of the overflow probability mentioned in Chapter II.

A. BEHAVIOR OF DISTRIBUTION OF COMPUTATION

The computation performed by the Fano decoder is a random variable. It is large during periods of high channel noise and small otherwise. The same is true of the random variable of "static" computation C associated with the q^{th} node of the correct path. We now argue somewhat loosely that exponential growth of "static" computation implies that it has an algebraic distribution function.

Let $\bar{\xi}_s$ be the sequence of $s\ell$ channel transitions (corresponding to s tree branches) following the q^{th} correct node. The sequence $\bar{\xi}_s$ alone is not sufficient, as a rule, to determine C completely. Knowledge of $\bar{\xi}_s$ is sufficient, however, to determine whether C is large or not. If $\bar{\xi}_s$ for large s represents a long interval of high channel noise, then C will still be random, but all values in its range of values will be very large. In particular, let us assume that for each $s \geqslant s_o$ there exists a $\bar{\xi}_s$ such that $C \geqslant A_o 2^{s\Theta}$ where A_o, $\Theta > 0$, that is, the "static" computation grows exponentially with the length of an interval of high channel noise. (Following arguments similar to those of the next section, it may be verified that such an assumption holds for all codes on the completely connected DMC.)

$$P_R\,[C \geqslant L] \geqslant P_R\,[C \geqslant L|\bar{\xi}_s]\; P_R\,[\bar{\xi}_s] \tag{10}$$

where $P_R\,[\bar{\xi}_s]$ is the probability that the particular sequence $\bar{\xi}_s$ of $s\ell$ channel transitions is the sequence of $s\ell$ transitions following the q^{th} reference node. Both $\bar{\xi}_s$ and s in Eq. (10) are arbitrary. For each s let $\bar{\xi}_s$ be a high channel noise sequence. Now choose s such that

$$A_o 2^{s\Theta} \geqslant L > A_o 2^{(s-1)\Theta} \quad . \tag{11}$$

Then, for this s and the high channel noise sequence $\bar{\xi}_s$ we have by assumption that $C \geqslant A_o 2^{s\Theta}$. Therefore, from Eq. (11), $C \geqslant L$ which implies that $P_R\,[C \geqslant L|\bar{\xi}_s] = 1$. Thus, for the particular value of s defined by Eq. (11) and for the high channel noise sequence $\bar{\xi}_s$ of that length, we have

$$P_R\,[C \geqslant L] \geqslant P_R\,[\bar{\xi}_s] \quad . \tag{12}$$

For the completely connected DMC (the only channels considered in this chapter) we have

$$P_R\,[\bar{\xi}_s] \geqslant 2^{-s\varphi} \tag{13}$$

where $\varphi \triangleq -\ell \log_2 \min_{j,k} p\,[y_j/y_k]$ because $P_R\,[\bar{\xi}_s]$ is the product of $s\ell$ channel transition probabilities all of which exceed the smallest transition probability, the latter being nonzero by the connectedness assumption. Combining Eqs. (11) and (13) we have the following lower bound to $P_R\,[C \geqslant L]$. The bound applies only for $s \geqslant s_o$ or $L \geqslant A_o 2^{s_o\Theta}$.

$$P_R\,[C \geqslant L] \geqslant \left(\frac{A_o}{L}\right)^{\varphi/\Theta} 2^{-\varphi} \quad \text{for} \quad L \geqslant L_o \triangleq A_o 2^{s_o\Theta} \quad . \tag{14}$$

Exponential growth of computation with the length of an interval of high channel noise implies that the distribution of "static" computation is algebraic, which in turn implies the particular sensitivities of the overflow probability discussed in Chapter II. The existence of exponential growth is, therefore, a most important characteristic (or defect) of a decoding scheme.

B. LOWER BOUND ARGUMENT

Our intention in this section is to underbound, without a loss of rigor, the probability $P_R\,[C \geqslant L]$. To underbound $P_R\,[C \geqslant L]$, we find an event which implies that $C \geqslant L$. The probability of the former event underbounds the probability that $C \geqslant L$ and is used as the underbound to $P_R\,[C \geqslant L]$. As a preliminary, we recall some of the definitions and statements of Chapter II.

"Static" computation associated with the q^{th} incorrect subset is defined as the number of forward or backward "looks" required by the Fano decoder on the reference node (the q^{th} correct node) or on nodes in the q^{th} incorrect subset. "Static" computation is measured under the assumption that the decoder decodes without error, that the q^{th} correct node is in the infinite past of the decoding process, and that the buffer has infinite storage capacity. The latter assumption is equivalent to the assumption that the machine can search forward or backward to any length in the tree.

A node in the q^{th} incorrect subset is labeled (m, s, q) to indicate that it is at penetration s in this subset (there are s branches between it and the reference node) and it is m^{th} in order among the M(s) nodes at that penetration in the q^{th} incorrect subset; M(s) is given below.

$$M(0) = 1$$

$$M(s) = (b-1)\,b^{s-1} \quad \text{for} \quad s \geqslant 1 \quad . \qquad \text{[Eq. (1)]}$$

There are b^t nodes at penetration t or less, since

$$\begin{aligned}\sum_{s=0}^{t} M(s) &= 1 + (b-1) + (b-1)\,b + \ldots + (b-1)\,b^{t-1} \\ &= 1 + (b-1)\,(1 + b + b^2 + \ldots + b^{t-1}) \\ &= 1 + (b-1)\left(\frac{b^t-1}{b-1}\right) = b^t \quad .\end{aligned} \tag{15}$$

The reference node is labeled (1, 0, q) and is said to be at penetration zero in the q^{th} incorrect subset.

A path metric is defined and the value of the path metric on a path terminated by node (m, s, q) is associated with node (m, s, q) and is called d(m, s, q). Let $\bar{u}_n$ represent the path of $n = q + s$ branches terminated by (m, s, q) and let $\bar{v}_n$ be the corresponding portion of the received sequence.† Then, d(m, s, q) is defined as

$$d(m, s, q) \triangleq \sum_{r=1}^{n} \sum_{h=1}^{\ell} [I(u_{rh}, v_{rh}) - R] \qquad \text{[Eq. (7)]}$$

whcre

$$I(u_{rh}, v_{rh}) \triangleq \log_2 \frac{p\,[v_{rh}/u_{rh}]}{f(v_{rh})} \qquad \text{[Eq. (6)]}$$

and u_{rh}, v_{rh} arc thc h^{th} of ℓ digits on the r^{th} branches of $\bar{u}_n$, respectively. $p\,[v_{rh}/u_{rh}]$ is a channel transition probability and $f(v_{rh})$ is a probability-like function which is interpreted as the probability of the channel output digit v_{rh} when channcl inputs are assigned probabilitics $\{p_k\}$, $1 \leqslant k \leqslant K$.

As the Fano decoder operates, it attempts to extend along a path which increases in path metric. A set of threshold $T_i = i\,t_o$, $-\infty < i < \infty$, is used to ascertain whether a path being examined grows or decreases in metric. The decoder operation depends on increments in the path metric. Thus, we may assume that the reference node (1, 0, q) lies between $T_o = 0$ and $T_1 = t_o$, i.e., $0 \leqslant d(1, 0, q) \leqslant t_o$.

Our intent is to find an event which implies that $C \geqslant L$ and to underbound the probability of this event. It was observed in Chapter II that if D is defined as the minimum value of the correct path metric at or following (1, 0, q), and T_D is the threshold just below D, then at least one computation (a forward look) is required on node (m, s, q) and on each node connecting it to (1, 0, q) if $D < 0$, and node (m, s, q) and all nodes connecting it to (1, 0, q) lie above $T_D + t_o$.

One forward look on node (m, s, q) and each of the connecting nodes is required under these conditions before the decoder reduces the running threshold from $T_D + t_o$ to T_D. This latter threshold is used at least once since the decoded path is the correct path (by assumption) and this path dips below $T_D + t_o$ (see Fig. 11).

We assume that the channel is completely connected. This implies that the path terminated by some node (m, t, q) cannot fall from the value of the metric on the reference node, d(1, 0, q), with a slope‡ of magnitude larger than $\ell(R - I_{min})$ where§

$$I_{min} \triangleq \min_{j,k} \log_2 \frac{p\,[y_j/x_k]}{f(y_j)} \quad . \qquad (16)$$

That is,

$$d(m, t, q) \geqslant d(1, 0, q) - t\ell(R - I_{min}) \quad . \qquad (17)$$

† The subscript n on $\bar{u}$ or $\bar{v}$ is reserved for sequences of n branches measured from the origin. The subscript s on $\bar{u}$ or $\bar{v}$ will indicote sequences of s bronches meosured from the q^{th} correct node.

‡ Slope is defined os the increment in the metric for o one-node chonge in path penetrotion.

§ It moy be shown thot $I_{min} \leqslant 0$.

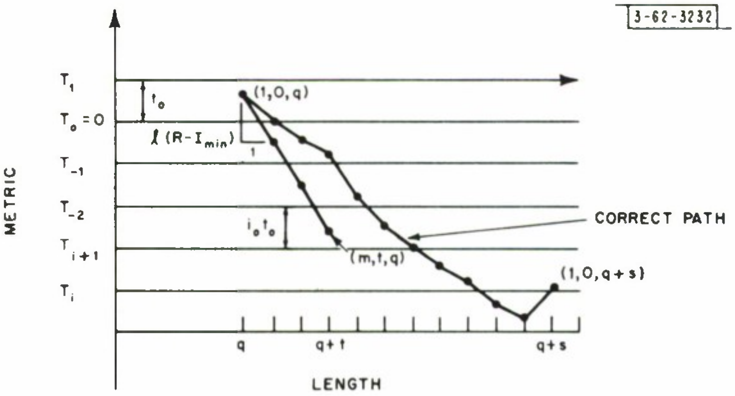


Fig. 11. Trajectories of correct path and incorrect path.

We are now prepared to describe an event which implies $C \geqslant L$. As shown by Eq. (15), there are b^t nodes at penetration t or less in the q^{th} incorrect subset. If each of these b^t nodes, $b^t \geqslant L > b^{t-1}$, lies above some threshold T_i, and if the correct path falls below T_i at some node beyond (1, 0, q), say at node (1, 0, q + s) [which is s branches removed from (1, 0, q)], we find that the "static" computation on just the b^t nodes of the incorrect subset equals or exceeds L (since t is defined by $b^t \geqslant L > b^{t-1}$) so that the total "static" computation C equals or exceeds L.

We have the desired underbound if we let T_i be the threshold below the value of the path metric on the path (m, t, q) which falls at the maximum rate. In particular, we have that

$$d(1, 0, q) - t\ell(R - I_{min}) \geqslant T_i > d(1, 0, q) - t\ell(R - I_{min}) - t_o \quad . \tag{18}$$

If the correct path falls below this underbound to T_i, then threshold T_i is used and at least b^t, $b^t \geqslant L > b^{t-1}$, nodes in the q^{th} incorrect subset will have at least b^t computations done on them. Therefore, the probability that the correct path falls below the T_i of Eq. (18) underbounds P_R $[C \geqslant L]$.

The metric on the $(q + s)^{th}$ correct node is defined as d(1, 0, q + s). If d(1, 0, q + s) is less than the underbound to T_i, this threshold will be used. This condition is written as

$$d(1, 0, q + s) \leqslant d(1, 0, q) - t\ell(R - I_{min}) - t_o \quad . \tag{19}$$

If we let $\bar{u}_s$ represent the s branches of the correct path which follow node (1, 0, q) and let $\bar{v}_s$ be the corresponding section of the received sequence, we have from Eqs. (6) and (7)

$$d(1, 0, q + s) - d(1, 0, q) = \sum_{r=1}^{s} \sum_{h=1}^{\ell} [I(u_{rh}, v_{rh}) - R] \tag{20}$$

$$d(1, 0, q + s) - d(1, 0, q) \triangleq I(\bar{u}_s, \bar{v}_s) - s\ell R \tag{21}$$

where u_{rh}, v_{rh} are the h^{th} digits on the r^{th} branches of $\bar{u}_s$, $\bar{v}_s$, respectively. Equation (19) is now rewritten with the aid of Eq. (21).

$$I(\bar{u}_s, \bar{v}_s) \leqslant s\ell R - t\ell(R - I_{min}) - t_o \quad . \tag{22}$$

Recalling that $b^t \geqslant L > b^{t-1}$ and remembering that $R \triangleq (\log_2 b)/\ell$, we obtain the final result, namely, if

$$I(\bar{u}_s, \bar{v}_s) \leqslant s\ell R - (\log_2 L + 1) \left(\frac{R - I_{min}}{R}\right) - t_o \tag{23}$$

then the static computation C must exceed L. Therefore, the probability of the event in Eq. (23) underbounds P_R [$C \geqslant L$]. We note that s is arbitrary. It is chosen to maximize the underbound to P_R [$C \geqslant L$]. The desired result then is

$$P_R\ [C \geqslant L] \geqslant \max_s\ P_R \left[I(\bar{u}_s, \bar{v}_s) \leqslant s\ell R - t_o - (\log_2 L + 1) \left(\frac{R - I_{min}}{R}\right)\right] \quad . \tag{24}$$

It should be noted that the random variable $I(\bar{u}_s, \bar{v}_s)$ is assigned with probability $P_R\ [\bar{u}_s, \bar{v}_s]$, which is the probability that the first s branches of the transmitted and received sequences following the q^{th} correct node are $\bar{u}_s, \bar{v}_s$, respectively. The inequality of Eq. (24) applies to any particular code and $\bar{u}_s$ is a codeword (of s branches) in this code.

Let $\rho_s(x) \triangleq P_R\ [I(\bar{u}_s, \bar{v}_s) \leqslant x]$. Then, the lower bound result is formally summarized below.

Theorem 1.

The "static" computation in the q^{th} incorrect subset, when the Fano algorithm is used on the completely connected DMC, has the following bound on its cumulative probability distribution:

$$P_R\ [C \geqslant L] \geqslant \max_s\ \rho_s \left[s\ell R - t_o - (\log_2 L + 1) \left(\frac{R - I_{min}}{R}\right)\right] \quad . \tag{25}$$

where I_{min} is defined by Eq. (16).

Next we further lower bound Eq. (25) so that the dependence of the bound on L and R becomes explicit. First, we lower bound $\rho_s(x)$ in terms of the smallest value of the conditional probability $\rho_s(x|\bar{u}_s)$, defined as the conditional probability that $I(\bar{u}_s, \bar{v}_s) \leqslant x$ given $\bar{u}_s$.

$$\rho_s(x) \triangleq \sum_{\bar{u}_s \text{ in the code}} \rho_s(x|\bar{u}_s)\ P_R\ [\bar{u}_s] \tag{26}$$

$$\rho_s(x) \geqslant \min_{\text{all } \bar{u}_s}\ \rho_s(x|\bar{u}_s) \quad . \tag{27}$$

Here the minimum is taken over all words of $s\ell$ digits, not just words in the code. Since Eq. (27) is independent of code, we shall use it to obtain a bound valid for all codes. Equality holds in Eq. (27) under certain conditions on the channel and the probability-like function $f(y_j)$. Equality is equivalent to saying that $\rho_s(x)$ is independent of the code. The conditions are:

(a) The channel is uniform at the input, i.e., the set of transition probabilities $\{p(y_j/x_k)\}$, $1 \leqslant j \leqslant J$ is independent of k;

(b) $f(y_j)$ = constant for all $1 \leqslant j \leqslant J$.

In the second major step directed at exhibiting the dependence of the bound on L and R, we introduce and apply a theorem due to Gallager.[17] We shall use it to underbound $\rho_s(x|\bar{u}_s)$. Although it is a weaker theorem than the Central Limit Theorem for Large Deviations (Ref. 18), it is sufficient to demonstrate the dependencies of $P_R\,[C \geqslant L]$ for large L because the two theorems are asymptotically equal.

Theorem 2. (Gallager)

Let $\{\xi_i\}$, $1 \leqslant i \leqslant N$ be a set of statistically independent random variables. ξ_i assumes the J values w_{ij}, $1 \leqslant j \leqslant J$, with probabilities $\{P_R(w_{ij})\}$. Let ξ be the sum of these N variables, $\xi = \sum_{i=1}^{N} \xi_i$. Define $\mu_i(\sigma)$ by†

$$\mu_i(\sigma) \triangleq \log_2 \overline{2^{\sigma\xi_i}} \quad . \tag{28}$$

Then,

$$\mu(\sigma) \triangleq \log_2 \overline{2^{\sigma\xi}} = \sum_{i=1}^{N} \mu_i(\sigma) \tag{29}$$

and for $\sigma \leqslant 0$ we have

$$P_R\,[\xi \leqslant \mu'(\sigma)] \geqslant \frac{1}{2}\, 2^{[\mu(\sigma)-\sigma_{\mu'}(\sigma)]} \exp\left[-\frac{4}{e}\sqrt{\frac{2N(1-p_{\min})}{p_{\min}}}\right] \tag{30}$$

where the prime indicates differentiation with respect to σ, and $p_{\min}$ is defined by

$$p_{\min} \triangleq \min_i P_R\,[\min_j w_{ij}] \quad . \tag{31}$$

To use this theorem in underbounding $\rho_s(x|\bar{u}_s)$, we must associate the N random variables $\{\xi_i\}$ with the random variables appearing in the definition of $\rho_s(x|\bar{u}_s)$. We recall that

$$\rho_s(x|\bar{u}_s) = P_R\,[I(\bar{u}_s, \bar{v}_s) \leqslant x|\bar{u}_s] \tag{32}$$

where $I(\bar{u}_s, \bar{v}_s)$ is defined from Eqs. (20) and (21) as

$$I(\bar{u}_s, \bar{v}_s) = \sum_{r=1}^{s} \sum_{h=1}^{\ell} \log_2 \frac{p\,[v_{rh}/u_{rh}]}{f(v_{rh})} \tag{33}$$

and u_{rh}, v_{rh} are the h^{th} of ℓ digits on the r^{th} branches of $\bar{u}_s$, $\bar{v}_s$, respectively. With $\bar{u}_s$ fixed, this random variable $I(\bar{u}_s, \bar{v}_s)$ is assigned with probability

$$P_R\,[\bar{v}_s|\bar{u}_s] = \prod_{r=1}^{s} \prod_{h=1}^{\ell} p\,[v_{rh}/u_{rh}] \quad . \qquad \text{[Eq. (2)]}$$

The $s\ell$ random variables

$$\left\{\log_2 \frac{p\,[v_{rh}/u_{rh}]}{f(v_{rh})}\right\}$$

† The bar notation $\overline{\qquad}$ indicates a statistical average.

are therefore statistically independent and assigned with probabilities $p\,[v_{rh}/u_{rh}]$. Thus, if we make the following indentifications, Theorem 2 applies to $\rho_s(x|\bar{u}_s)$:

$$
\begin{aligned}
N &\triangleq s\ell \\
i &\triangleq (r-1)\,\ell + h \\
\xi_i &\triangleq \log_2 \frac{p\,[v_{rh}/u_{rh}]}{f(v_{rh})} \\
w_{ij} &\triangleq \log_2 \frac{p\,[y_j/u_{rh}]}{f(y_j)} \\
P_R\,[w_{ij}] &\triangleq p\,[y_j/u_{rh}] \\
\mu'(\sigma) &\triangleq x \quad .
\end{aligned}
\tag{34}
$$

The particular definition of the index i is one which leads to a natural ordering of the $s\ell$ pairs (u_{rh}, v_{rh}).

Before we apply Theorem 2 to $\rho_s(x|\bar{u}_s)$ we observe that by decreasing p_{min} we further weaken the inequality of Eq. (30). Therefore, we may replace p_{min} with P_{min},

$$P_{min} \triangleq \min_{j,\,k}\; p\,[y_j/x_k] \quad . \tag{35}$$

Now let us consider the form of $\mu_i(\sigma)$ and of $\mu(\sigma)$. From the definitions of Eq. (34) we have

$$\mu_i(\sigma) = \log_2 \sum_{j=1}^{J} p\,[y_j/u_{rh}]^{1+\sigma}\, f(y_j)^{-\sigma} \quad . \tag{36}$$

If we define $\underline{Q}_o = (q_1, \ldots, q_k)$ as the composition of codeword $\bar{u}_s$, that is, if Nq_k represents the number of times channel input symbol x_k appears in $\bar{u}_s$, $\sum_{k=1}^{K} q_k = 1$, then we have for $\mu(\sigma)$ the following:

$$\mu(\sigma) = \sum_{i=1}^{N} \mu_i(\sigma) = N \sum_{k=1}^{K} q_k \gamma_k(\sigma) \tag{37}$$

where

$$\gamma_k(\sigma) = \log_2 \sum_{j=1}^{J} p\,[y_j/x_k]^{1+\sigma}\, f(y_j)^{-\sigma} \quad . \tag{38}$$

All terms of Theorem 2 have been defined so that we may now state the desired lower bound to $\rho_s(x|\bar{u}_s)$. If $\bar{u}_s$ has composition $\underline{Q}_o$, then,

$$\rho_s(x|\bar{u}_s) \geqslant \frac{1}{2}\, 2^{N \sum_{k=1}^{K} q_k[\gamma_k(\sigma) - \sigma\gamma_k'(\sigma)]} \exp\left[-\frac{4}{e}\sqrt{\frac{2N(1-P_{min})}{P_{min}}}\right] \quad . \tag{39}$$

This bound is independent of the order of symbols in the codeword. Therefore, for that (unusual) class of codes having all codewords of the same composition, this lower bound applies directly to all words $\bar{u}_s$ in the code. Moreover, independence of the order of symbols in a codeword applies to $\rho(x|\bar{u}_s)$ as well as to its lower bound: it can be shown that $\rho_s(x|\bar{u}_s') = \rho_s(x|\bar{u}_s)$ when $\bar{u}_s'$ and $\bar{u}_s$ have the same composition. It follows that the inequality of Eq. (27) is weaker than necessary for codes of fixed and known composition; for this class of codes we may write

$$\rho_s(x) = \rho_s(x|\bar{u}_s) \tag{40}$$

for any $\bar{u}_s$ in the code. It should be noted again that Eq. (40) applies only to codes of fixed composition, whereas Eq. (27) applies to all codes.

Our primary task is to exhibit the dependence of the bound of Theorem 1 on L and R. We now have the necessary tools to do this. We use either Eq. (40) or Eq. (27), depending on whether the bound is to apply to a code of fixed composition or is to apply to all codes, together with the bound of Eq. (39) and the inequality of Eq. (25) of Theorem 1. We shall consider the fixed composition case first since it serves as an introduction to the general lower bound.

For fixed $\underline{Q}_o$, we have from Theorem 1, the definition of Eq. (34), the equivalence of the statement in Eq. (40), Theorem 2 and the bound of Eq. (39) the following lower bound to $P_R\,[C \geqslant L]$:

$$P_R\,[C \geqslant L] \geqslant \max_N \rho_s(x) \geqslant \max_N \rho_s(x|\bar{u}_s)$$

$$\geqslant \frac{1}{2} \max_N \left\{ 2^{N \sum_{k=1}^{K} q_k[\gamma_k(\sigma)-\sigma\gamma_k'(\sigma)]} \exp\left[-\frac{4}{e}\sqrt{\frac{2N(1-P_{min})}{P_{min}}}\right]\right\} \tag{41}$$

where

$$\sigma \leqslant 0$$

$$x = NR - F$$

$$F \triangleq t_o + (\log_2 L + 1)\left(\frac{R - I_{min}}{R}\right)$$

$$N \triangleq s\ell \quad . \tag{42}$$

The maximization over N in Eq. (41) is taken subject to the following constraint

$$\sum_{k=1}^{K} q_k\gamma_k'(\sigma) = R - \frac{F}{N}$$

or

$$N = \frac{F}{R - \sum\limits_{k=1}^{K} q_k\gamma_k'(\sigma)} \tag{43}$$

which is implied by the first equation in Eq. (42), the last equation in Eq. (34) and the definition of $\mu(\sigma)$, Eq. (37). The function F is independent of $\underline{Q}_o$ and N, and is constant with respect to the maximization.

Strictly speaking, the maximization on N must be taken only for values of N which are multiples of ℓ, the number of digits per tree branch. We now drop this constraint and permit N to assume all values $1 \leqslant N < \infty$. The imprecision introduced neither affects the character of the end result nor materially alters its numerical value.

Let us now consider the connection between N and σ from the second equation in Eq. (43). One can show that

$$\gamma_k''(\sigma) = \overline{(\xi_i' - \bar{\xi}_i')^2} \geqslant 0 \tag{44}$$

where ξ_i' assumes the same values as does ξ_i of Eq. (34) but it is assigned each such value with the probability

$$P_R\,[\xi_i' = w_{ij}] = \frac{p\,[y_j/u_{rh}]^{1+\sigma}\; f(y_j)^{-\sigma}}{\sum\limits_{j=1}^{J} p\,[y_j/u_{rh}]^{1+\sigma}\; f(y_j)^{-\sigma}} \tag{45}$$

when $u_{rh} = x_k$. Consequently, $\gamma_k'(\sigma)$ is monotone increasing in σ, which implies that N is monotone increasing in σ. Since $0 \leqslant N < \infty$, we must restrict σ in Eq. (43) to be less than the value at which N is infinite. We shall impose this restriction implicitly by extending the definition of $1/[R - \Sigma\; q_k\gamma_k'(\sigma)]$ so that it is infinite for σ larger than the critical value. At the end of the next paragraph, it will become clear that this extension does not affect the maximization, serving only to simplify the analysis.

We return now to the maximization of Eq. (37). If h(N) and q(N) are positive, then

$$\max_N\; h(N)\, q(N) \geqslant [\max_N\; h(N)]\; q(N') \tag{46}$$

where N' may assume any value. Thus, if we maximize Eq. (41) with respect to the first of the two factors, we further lower bound $P_R\,[C \geqslant L]$. The maximum of the first factor occurs at the maximum of the exponent

$$N\epsilon(\sigma) \triangleq N \sum_{h=1}^{K} q_k[\gamma_k(\sigma) - \sigma\gamma_k'(\sigma)] \quad . \tag{47}$$

Let us study this exponent. It is negative since $\epsilon(\sigma)$ is negative. We see this by observing that $\epsilon(\sigma)$ assumes value zero at $\sigma = 0$ and has derivative $\sum\limits_{k=1}^{K} q_k(-\sigma)\; \gamma_k''(\sigma) \geqslant 0$ for $\sigma \leqslant 0$, the range of σ of interest. To determine whether the exponent $N\epsilon(\sigma)$ has a maximum in σ, we take the first derivative with respect to σ.

$$\frac{d}{d\sigma} N \sum_{k=1}^{K} q_k\,[\gamma_k(\sigma) - \sigma\gamma_k'(\sigma)] = F\,\frac{d}{d\sigma}\,\frac{\sum\limits_{k=1}^{K} q_k\,[\gamma_k(\sigma) - \sigma\gamma_k'(\sigma)]}{R - \sum\limits_{k=1}^{K} q_k\gamma_k'(\sigma)}$$

$$= \frac{(-\sigma)\left[\sum\limits_{k=1}^{K} q_k\gamma_k''(\sigma)\right]\left[R - \sum\limits_{k=1}^{K} q_k\,\frac{\gamma_k(\sigma)}{\sigma}\right]}{\left[R - \sum\limits_{k=1}^{K} q_k\gamma_k'(\sigma)\right]^2} \quad . \tag{48}$$

All factors are positive for $\sigma \leqslant 0$, with the possible exception of the term $R - \sum_{k=1}^{K} q_k\, [\gamma_k(\sigma)]/\sigma$. Since $\sum_{k=1}^{K} q_k\, [\gamma_k(\sigma)]/\sigma$ has derivative

$$\frac{d}{d\sigma} \sum_{k=1}^{K} q_k \frac{\gamma_k(\sigma)}{\sigma} = \sum_{k=1}^{K} q_k \left[\frac{\sigma\gamma_k'(\sigma) - \gamma_k(\sigma)}{\sigma^2}\right] = -\frac{\epsilon(\sigma)}{\sigma^2} \geqslant 0 \tag{49}$$

we find that $R - \sum_{k=1}^{K} q_k\, [\gamma_k(\sigma)]/\sigma$ is positive for $\sigma \leqslant \sigma_o$ and negative for $\sigma \geqslant \sigma_o$ where σ_o is such that

$$R = \sum_{k=1}^{K} q_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \quad . \tag{50}$$

We can now sketch $N\epsilon(\sigma)/F$ for $\sigma \leqslant 0$ (see Fig. 12). It is negative for $\sigma \leqslant 0$ and has a maximum at $\sigma = \sigma_o$. The value of this maximum is

$$\frac{N(\sigma_o)\ \epsilon(\sigma_o)}{F} = \frac{\epsilon(\sigma_o)}{\frac{1}{\sigma_o}\cdot\left(\sum_{k=1}^{K} q_k\, [\gamma_k(\sigma_o) - \sigma_o\gamma_k'(\sigma_o)]\right)} = \sigma_o \tag{51}$$

i.e., the maximum (σ_o, σ_o) lies on a straight line of slope one passing through the origin. For $\sigma \leqslant \sigma_o$, $R \geqslant \sum_{k=1}^{K} q_k\, [\gamma_k(\sigma_o)/\sigma_o]$ so that $N\epsilon(\sigma)/F \geqslant \sigma$, that is, $N\epsilon(\sigma)/F$ lies above the unit slope line passing through the origin for $\sigma \leqslant \sigma_o$. Maximizing $N\epsilon(\sigma)$ over N is equivalent to maximizing this exponent over σ where N and σ are related by Eq. (43). Therefore, the maximum of the first term in Eq. (41), $2^{\sigma_o F}$, is related parametrically to the rate R by Eq. (50).

The final bound is obtained if in the second factor of Eq. (41) we use $N' = N(\sigma_o)$, the value of N which maximizes the first factor. Then using Eq. (46) we have for the fixed composition case

$$P_R\,[C \geqslant L] \geqslant \frac{1}{2}\, 2^{\sigma_o F} \exp\left[-\frac{4}{e}\sqrt{F}\sqrt{2\,\frac{\sigma_o}{\epsilon(\sigma_o)}\,\frac{1 - P_{min}}{P_{min}}}\right] \tag{52}$$

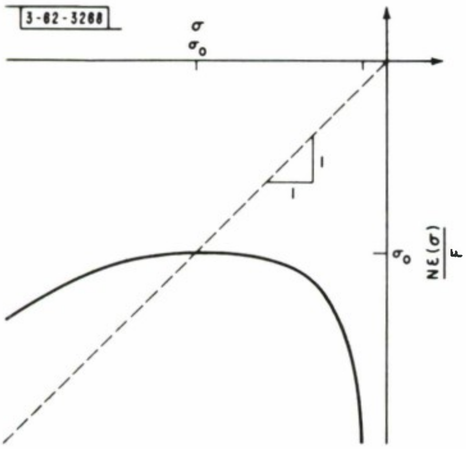


Fig. 12. Behavior of $N\epsilon(\sigma)/(F)$ with σ.

where $\sigma_o \leqslant 0$ is such that

$$R = \sum_{k=1}^{K} q_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \quad . \qquad \text{[Eq. (50)]}$$

The range $-1 \leqslant \sigma_o \leqslant 0$ suffices since, as shown in Eq. (49), the sum in Eq. (50) is monotone increasing in σ, being negative for $\sigma < -1$. This is the lower bound result for the fixed composition case. We must now consider the general lower bound, valid for all codes. We shall use many of the results obtained above.

To obtain the general lower bound, we lower bound $P_R\,[C \geqslant L]$ using Theorem 1 and inequalities (27) and (39).

$$P_R\,[C \geqslant L] \geqslant \frac{1}{2} \max_{N} \min_{\underline{Q}_o} \left\{ 2^{N\epsilon(\sigma)} \exp\left[-\frac{4}{e}\sqrt{\frac{2N(1-P_{min})}{P_{min}}}\right]\right\} \tag{53}$$

where $\sigma \leqslant 0$. We would like to focus attention on the first of the two factors above. We justify our doing this as follows: Let $h(N, \underline{Q}_o)$, $g(N, \underline{Q}_o) \geqslant 0$. Then,

$$h(N, \underline{Q}_o) \geqslant \left\{\min_{\underline{Q}_o} h(N, \underline{Q}_o)\right\} \quad , \qquad g(N, \underline{Q}_o) \geqslant \left\{\min_{\underline{Q}_o} g(N, \underline{Q}_o)\right\}$$

$$h(n, \underline{Q}_o)\, g(N, \underline{Q}_o) \geqslant \left\{\min_{\underline{Q}_o} h(N, \underline{Q}_o)\right\} \left\{\min_{\underline{Q}_o} g(N, \underline{Q}_o)\right\}$$

so that

$$\min_{\underline{Q}_o} \{h(N, \underline{Q}_o)\, g(N, \underline{Q}_o)\} \geqslant \min_{\underline{Q}_o} \{h(N, \underline{Q}_o)\} \min_{\underline{Q}_o} \{g(N, \underline{Q}_o)\}$$

and

$$\max_{N} \min_{\underline{Q}_o} \{h(N, \underline{Q}_o)\, g(N, \underline{Q}_o)\} \geqslant \max_{N} \min_{\underline{Q}_o} \{h(N, \underline{Q}_o)\} \min_{\underline{Q}_o} \{g(N', \underline{Q}_o)\} \quad . \tag{54}$$

In the last step we have used Eq. (46). Thus, if we minimize the second term in Eq. (53) on $\underline{Q}_o$ and use in it the value of σ which achieves the max-min of the first term we will have a valid lower bound. We minimize the second factor $\underline{Q}_o$ if we maximize N' on $\underline{Q}_o$.

$$N_{max}(\sigma) \triangleq \max_{\underline{Q}_o} N'(\sigma) = \frac{F}{R - \max_{k} \gamma_k(\sigma)} \quad . \tag{55}$$

Then, we have

$$P_R\,[C \geqslant L] \geqslant \frac{1}{2} \left\{ 2^{\max_{N} \min_{\underline{Q}_o} N\epsilon(\sigma)} \right\} \exp\left[-\frac{4}{e}\sqrt{\frac{2N_{max}(\sigma)\,(1-P_{min})}{P_{min}}}\right] \quad . \tag{56}$$

Our next concern is with the max-min of $N\epsilon(\sigma)$. We assert that the minimum on $\underline{Q}_o$ (the components of $\underline{Q}_o$ are positive and sum to one) of $N\epsilon(\sigma)$ occurs when $\underline{Q}_o$ has a single nonzero component, having the value unity. This component $q_{k_o} = 1$ is such that fixed $\sigma \leqslant 0$

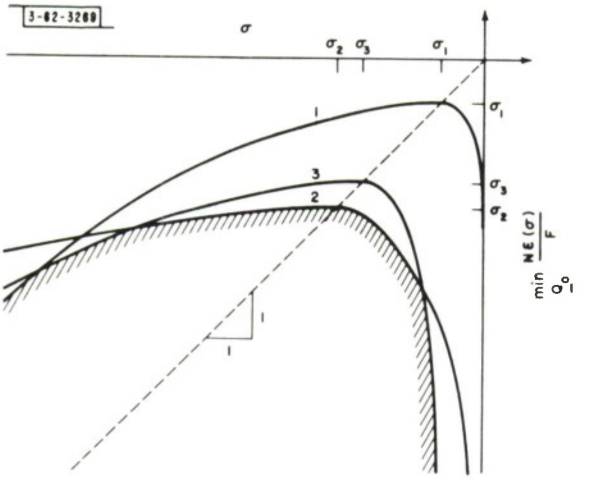


Fig. 13. Minimizotion of $N\epsilon(\sigma)/(F)$ over $\underline{Q}_o$.

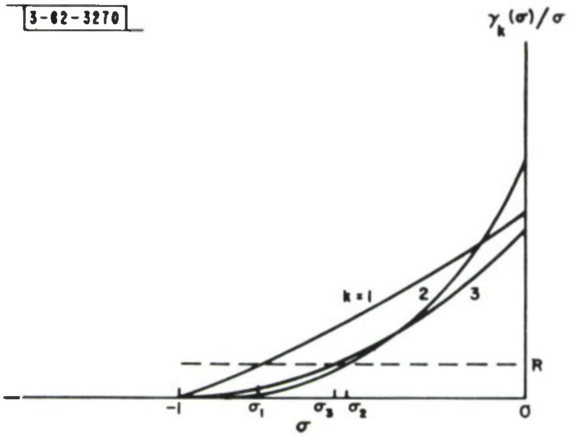


Fig. 14. Relotive volues of $\gamma_k(\sigma)/\sigma$.

$$\frac{\gamma_{k_o}(\sigma) - \sigma\gamma'_{k_o}(\sigma)}{R - \gamma'_{k_o}(\sigma)} \leqslant \frac{\gamma_k(\sigma) - \sigma\gamma_k(\sigma)}{R - \gamma_k(\sigma)} \qquad \text{all } k \quad . \tag{57}$$

This assertion is proved as follows: Let δ be defined as the difference between $N\epsilon(\sigma)/F$ for arbitrary $\underline{Q}_o$ and the value of $N\epsilon(\sigma)/F$ at the supposed minimum on $\underline{Q}_o$. Then we have

$$\delta \triangleq \sum_{k=1}^{K} \frac{q_k\,[\gamma_k(\sigma) - \sigma\gamma'_k(\sigma)]}{R - \sum\limits_{k=1}^{K} q_k\gamma'_k(\sigma)} - \frac{\gamma_{k_o}(\sigma) - \sigma\gamma'_{k_o}(\sigma)}{R - \gamma'_{k_o}(\sigma)} \quad . \tag{58}$$

Using Eq. (57) and remembering that by extension of its definition $R - \sum\limits_{k=1}^{K} q_k\gamma'_k(\sigma)$ cannot be negative for any $\underline{Q}_o$, we see that $\delta \geqslant 0$. We also observe that $\delta = 0$ for the assumed composition. Thus, this composition achieves the minimum. Now if we sketch $[\gamma_k(\sigma) - \sigma\gamma'_k(\sigma)]/[R - \gamma_k(\sigma)]$ for each k and all values of $\sigma \leqslant 0$ (keeping in mind that $R - \gamma'_k(\sigma) \geqslant 0$ by extension of its definition) we see that we achieve the minimum $N\epsilon(\sigma)$ on $\underline{Q}_o$ by taking the lower envelope of these functions (see Fig. 13). Notice that the maxima of the individual functions occur on the straight line of unit slope passing through the origin. The maximum of the k^{th} function occurs at $\sigma = \sigma_k$ where σ_k is such that $R = \gamma_k(\sigma)/\sigma_k$. For $\sigma \leqslant \sigma_k$, the k^{th} function lies above the unit slope straight line passing through the origin.

Figure 13 provides a graphical interpretation of the function $\min\limits_{\underline{Q}_o} N\epsilon(\sigma)$ vs σ. We now concentrate on maximizing this minimum on N or, equivalently, on $\sigma \leqslant 0$. We assert that this maximum occurs in Fig. 13 on the straight line of unit slope. This should be clear from the figure. If σ_k is such that $R = \gamma_k(\sigma_k)/\sigma_k$, that is, if $\{\sigma_k\}$ are the loci of the maxima, we further assert that the maximum over σ of $\min\limits_{\underline{Q}_o} N\epsilon(\sigma)$ occurs for σ equal to the smallest of the σ_k. This too should be clear from the figure.

We have found that the max-min of the exponent $N\epsilon(\sigma)$ occurs at the maximum σ_k of one of the functions $[\gamma_k(\sigma) - \sigma\gamma'_k(\sigma)]/R - \gamma'_k(\sigma)$, and that this particular maximum is the smallest of the maxima. At the particular maximum we have

$$\max_{N}\ \min_{\underline{Q}_o}\ N\epsilon(\sigma) = \sigma_o F \tag{59}$$

where σ_o is the smallest of the $\{\sigma_k\}$ satisfying $R = \gamma_k(\sigma_k)/\sigma_k$. Since $\gamma_k(\sigma)/\sigma$ is monotone increasing in σ from Eq. (55), we see from Fig. 14 that the smallest σ_k, as a function of R is the solution to the equation:

$$R = \max_{k} \frac{\gamma_k(\sigma_o)}{\sigma_o} \quad . \tag{60}$$

If we now choose $\sigma = \sigma_o$ in $N_{max}(\sigma)$, the value of N' in the exponent of the second factor of Eq. (56), we have

$$N_{max}(\sigma_o) = \frac{F}{\max\limits_{k} \frac{\gamma_k(\sigma_o)}{\sigma_o} - \max\limits_{k} \gamma'_k(\sigma_o)} \quad . \tag{61}$$

The denominator is positive because $\gamma_k(\sigma)/\sigma > \gamma'_k(\sigma)$ as implied by the fact that $\gamma_k(\sigma) - \sigma\gamma'_k(\sigma) = \epsilon(\sigma) < 0$, for $\sigma < 0$.

The complete general lower bound to $P_R [C \geqslant L]$, valid for all codes, can now be stated.

$$P_R [C \geqslant L] \geqslant \frac{1}{2} 2^{F\sigma_o} \exp\left\{-\frac{4}{e}\sqrt{F}\sqrt{\frac{2(1-P_{min})}{P_{min}\left[\max\limits_k \frac{\gamma_k(\sigma_o)}{\sigma_o} - \max\limits_k \gamma'_k(\sigma_o)\right]}}\right\} \tag{62}$$

where σ_o is the solution to the equation

$$R = \max_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \qquad . \qquad [\text{Eq. } (60)]$$

We collect the lower bounds to $P_R [C \geqslant L]$ for the two cases in the following theorem.

Theorem 3.

On the completely connected DMC, the random variable of "static" computation C has the following lower bound to its cumulative probability distribution function, $P_R [C \geqslant L]$:

$$P_R [C \geqslant L] \geqslant \frac{1}{2} 2^{F\sigma_o} \exp\left[-\frac{4}{e} A(\sigma_o) \sqrt{\frac{F(1-P_{min})}{P_{min}}}\right] \tag{63}$$

where

$$P_{min} \triangleq \min_{j,k} p [y_j/x_k] \qquad [\text{Eq. } (35)]$$

$$F \triangleq t_o + (\log_2 L + 1)\left(\frac{R - I_{min}}{R}\right) \qquad [\text{Eq. } (42)]$$

$$I_{min} \triangleq \min_{j,k} \log_2 \frac{p [y_j/x_k]}{f(y_j)} \qquad [\text{Eq. } (16)]$$

and $f(y_j)$ is a probability-like function of output symbol y_j, interpreted as the probability of y_j when channel inputs are assigned with probabilities $\{p_k\}$, $1 \leqslant k \leqslant K$.

$$f(y_j) \triangleq \sum_{k=1}^{K} p_k\, p [y_j/x_k] \quad . \tag{64}$$

The function $A(\sigma_o)$ and the parameter σ_o are related parametrically to the rate R. The relationship depends on whether the bound applies to all codes or to codes of known and fixed composition.

(1) For a code of fixed composition $\underline{Q}_o = (q_1, \ldots, q_k)$ we have

$$A(\sigma_o) \triangleq \left\{\frac{1}{\sqrt{\sum\limits_{k=1}^{K} q_k \left[\frac{\gamma_k(\sigma_o)}{\sigma_o} - \gamma_k(\sigma_o)\right]}}\right\} \tag{65}$$

$$R = \sum_{k=1}^{K} q_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \qquad \text{for} \quad -1 \leqslant \sigma_o \leqslant 0 \quad . \qquad [\text{Eq. }(50)]$$

(2) For all codes we may choose

$$A(\sigma_o) \triangleq \left[\frac{1}{\sqrt{\max\limits_k \frac{\gamma_k(\sigma_o)}{\sigma_o} - \max\limits_k \gamma_k'(\sigma_o)}} \right] \tag{66}$$

$$R = \max_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \qquad \text{for} \quad -1 \leqslant \sigma_o \leqslant 0 \quad . \qquad [\text{Eq. }(60)]$$

Here $\gamma_k(\sigma)$ is defined as

$$\gamma_k(\sigma) \triangleq \log_2 \sum_{j=1}^{J} p\,[y_j/x_k]^{1+\sigma}\, f(y_j)^{-\sigma} \quad . \qquad [\text{Eq. }(38)]$$

An important observation can be drawn immediately from the bound of Eq. (63). For very large F, corresponding to very large L, the bound is controlled almost entirely by the factor $2^{F\sigma_o}$. Thus, the bound behaves as $(1/L)^{(-\sigma_o)(R-I_{min})/R}$ for large L, so that the distribution is algebraic with large L.

CHAPTER IV
"RANDOM CODE" BOUND ON THE DISTRIBUTION OF COMPUTATION

The previous ehapter has established the algebraie eharacter of the distribution of "static" eomputation. In this chaptcr, we shall obtain an overbound to the distribution of computation averaged over the ensemblc of all tree codes. By so doing, we show that a largc number of codes exists whose distribution of "statie" computation is bounded by a multiple of the average. Together, the results of this chapter and of the preceding chapter de-limit the tail behavior of thc distribution of computation. Chapter V will interprct and relate the result of these two chapters.

A. RANDOM VARIABLE OF COMPUTATION

The approach we use to bound the ensemble average of the distribution of computation requircs that we overbound thc random variable of "static" computation. The discussion of Chapter II is sufficient to allow a bound on this random variable. We repeat thc pertinent arguments of that chapter.

"Static" computation associatcd with the q^{th} incorreet subset is defined as thc number of forward or backward "looks" requircd by the decoder in the incorreet subset assoeiated with the q^{th} node of the corrcct path. This subset eonsists of thc q^{th} corrcct node, labeled (1, 0, q), and of nodes on paths disjoint from that portion of the correct path whieh extends bcyond (1, 0, q). A particular nodc of this type is labeled (m, s, q) to indicate that it is in the q^{th} ineorrect subsct, is at "penetration" s, that is, is connected to (1, 0, q) through s branches, and is m^{th} in ordcr among the M(s) nodes at penetration s. The numbcr of nodes at penetration s, M(s), is defined below.

$$M(0) = 1$$

$$M(s) = (b-1)\,b^{s-1} \qquad \text{for } s \geqslant 1 \quad . \qquad \text{[Eq. (1)]}$$

The q^{th} corrcct node, or the reference node (1, 0, q) is said to be at penetration zero in the q^{th} incorrect subset.

A "path metric" d(m, s, q) on node (m, s, q) has been defined. If $\overline{\theta}$ is thc generic symbol rcpresenting the path terminating on node (m, s, q), then the path metric on this path of n = q + s branches is defined as follows:

$$d(m,s,q) = \sum_{r=1}^{n} \sum_{h=1}^{\ell} [I(\theta_{rh}, v_{rh}) - R] \tag{67}$$

where θ_{rh}, v_{rh} arc the h^{th} digits (of ℓ digits) on the r^{th} branehes of $\overline{\theta}$ and $\overline{v}_n$, the received sequence of n branches.† The function $I(\theta_{rh}, v_{rh})$ is defined by

$$I(\theta_{rh}, v_{rh}) = \log_2 \frac{P\,[v_{rh}/\theta_{rh}]}{f(v_{rh})} \tag{68}$$

where $f(v_{rh})$ is a probability-like funetion, interpreted as the probability of channel output symbol v_{rh} when channel inputs are assigned with probabilitics $\{p_k\}$, $1 \leqslant k \leqslant K$. That is, when $v_{rh} = y_j$, we have

† The subscript n on subsequences of the transmitted or received sequences, namely $\overline{u}_n$, $\overline{v}_n$, indicates their length in branches from the origin. The subscripts r, or s indicate their length from the reference node (1,0,q).

$$f(y_j) = \sum_{k=1}^{K} p_k P[y_j/x_k] \quad . \qquad \text{[Eq. (64)]}$$

Later in this chapter, we will find that $f(y_j)$ is equal to a probability appearing in the "random code" argument.

With this path metric, the Fano decoder searches paths in the tree code trying to find a path which tends to increase in path metric. A set of criteria $T_i = i\, t_o$ is defined. A path whosc path metric tends to cross an increasing sequence of criteria will with high probability be the correct path. As the machine searches for the correct path it must perform a number of forward or backward "looks" from nodes in the tree. We are concerned with a subset of the total computation ever performed, which consists of the number of computations eventually performed in the q^{th} incorrect subset. Since the machine computation depends on increments in the path metric, we may choose to let the value of the metric, $d(1, 0, q)$, on the first node of this subset, $(1, 0, q)$, lie between $T_o = 0$ and $T_1 = t_o$, that is, we may assume that $0 \leqslant d(1, 0, q) \leqslant t_o$.

We found in Chapter II that the computation in the q^{th} incorrect subset depends on the minimum value of the path metric at or following the reference node $(1, 0, q)$ and on the trajcctories of the individual incorrect paths. Let D be the correct path minimum at or following $(1, 0, q)$, and let T_D be the threshold just below D. We overbound computation on a particular node (m, s, q) by disregarding the history of the path preceding this node, looking only at thc value of the metric $d(m, s, q)$ on this particular node. If $d(m, s, q)$ is in a favorable position, we include node (m, s, q) in our computation count. As discussed in Chapter II, $d(m, s, q)$ is in a favorable position if $d(m, s, q) \geqslant T_D$. In particular, if $d(m, s, q) \geqslant T_k \geqslant T_D$, then the machine may do as many as $(b + 1)$ computations on node (m, s, q) with each such threshold T_k. If $T_k > d(m, s, q)$, the machine never does any computation on (m, s, q) with T_k.

Before we define a random variable which overbounds the random variable of "static" computation, we further consider the metric $d(m, s, q)$. Let $d(m, s)$ be the changc in $d(m, s, q)$ from thc value of the metric on the reference node, $d(1, 0, q)$. Then, if $\overline{\Theta}$ now reprcsents the s branches if the q^{th} incorrect subset preceding the node (m, s, q), and if $\overline{v}_s$ represents the corresponding portion of the receivcd sequence, we have

$$\begin{aligned} d(m, s) &\triangleq d(m, s, q) - d(1, 0, q) \\ &= I(\overline{\Theta}, \overline{v}_s) - s\ell R \triangleq \sum_{r=1}^{s} \sum_{h=1}^{\ell} [I(\Theta_{rh}, v_{rh}) - R] \end{aligned} \qquad (69)$$

where $I(\Theta_{rh}, v_{rh})$ is defined by Eq. (68). Then, since we have assumed that $d(1, 0, q)$ lics betwecn $T_o = 0$ and $T_1 = t_o$, we have that $d(m, s, q) \leqslant d(m, s) + t_o$. If $d(m, s, q)$ is replaced with this larger value for each node (m, s, q) the computation required on nodcs $\{(m, s, q)\}$ is increased, becausc these nodes may be examined with a larger number of thresholds. (Thc correct path minimum D is not changed.) Now, if we decrease by an equal amount the value of the path metric on each correct node following the reference node, we further increase the computation on nodes $\{(m, s, q)\}$. If we let $\overline{u}_{r_o}$, $\overline{v}_{r_o}$, bc the r_o branches of thc transmitted and received sequences following the reference node, and define $d(\overline{u}_{r_o}, \overline{v}_{r_o})$ as the change in the valuc of the metric from $d(1, 0, q)$ to $d(1, 0, q + r_o)$, the value of the metric on $(q + r_o)^{th}$ correct nodc, we have

$$d(\bar{u}_{r_o}, \bar{v}_{r_o}) \triangleq d(1, 0, q + r_o) - d(1, 0, q)$$

$$= I(\bar{u}_{r_o}, \bar{v}_{r_o}) - r_o \ell R \triangleq \sum_{r=1}^{r_o} \sum_{h=1}^{\ell} [I(u_{rh}, v_{rh}) - R] \quad . \tag{70}$$

We note that $d(1, 0, q) \geqslant 0$ so that $d(1, 0, q + r_o) \geqslant d(\bar{u}_{r_o}, \bar{v}_{r_o})$. If $d(1, 0, q + r_o)$ is replaced with $d(\bar{u}_{r_o}, \bar{v}_{r_o})$ computation on the incorrect nodes $\{(m, s, q)\}$ is increased. We are now prepared to an overbound to the random variable of "static" computation.

Using $d(m, s) + t_o$ for $d(m, s, q)$ and $d(\bar{u}_{r_o}, \bar{v}_{r_o})$ for $d(1, 0, q + r_o)$, $r_o \geqslant 0$, we raise the value of the metric on incorrect nodes and lower the value of the metric on correct nodes following the reference node. Thus, we overbound the computation on incorrect nodes. Equivalently, we overbound "static" computation. Now, as discussed above, the machine may do as many as $(b + 1)$ computations on node (m, s, q) with threshold T_k if $d(m, s) + t_o \geqslant T_k \geqslant T_{D'}$ where D' is the correct path minimum with the metric $d(\bar{u}_{r_o}, \bar{v}_{r_o})$. No computation is required on (m, s, q) with T_k if $d(m, s) + t_o < T_k$. Therefore, if there are N thresholds between $d(m, s) + t_o$ and $T_{D'}$, including $T_{D'}$, the machine may do as many as $(b + 1)$ N computations on node (m, s, q); N is a random variable. A convenient representation for N in terms of the upper bound to the value of the metric on node (m, s, q), $d(m, s) + t_o$, and the lower bound to the value of the metric on nodes of the correct path $d(\bar{u}_{r_o}, \bar{v}_{r_o})$ is had with the random variable $z_{i,s}(m)$. We define $z_{i,s}(m) = 1$ if $d(m, s) + t_o \geqslant T_i$ (that is, $d(m, s) \geqslant T_{i-1}$ since $T_i = i\, t_o$) and if $d(\bar{u}_{r_o}, \bar{v}_{r_o}) \leqslant T_{i+1}$ for some $r_o \geqslant 1$. If these conditions are not satisfied $z_{i,s}(m) = 0$. This type of random variable is called a characteristic function. Then,

$$z_{i,s}(m) = \begin{cases} 1 \text{ if } d(m, s) \geqslant T_{i-1} \text{ and } d(\bar{u}_{r_o}, \bar{v}_{r_o}) \leqslant T_{i+1} \text{ for some } r_o \geqslant 1 \\ \\ 0 \text{ otherwise} \quad . \end{cases} \tag{71}$$

A little reflection indicates that

$$\sum_{i=-\infty}^{\infty} z_{i,s}(m) = N$$

the number of thresholds between $d(m, s) + t_o$ and $T_{D'}$. Therefore,

$$(b + 1) \sum_{i=-\infty}^{\infty} z_{i,s}(m)$$

is an overbound to the computation on node (m, s, q). If this quantity is summed over all nodes in the q^{th} incorrect subset, that is, for $1 \leqslant m \leqslant M(s)$, $0 \leqslant s$, we have an overbound to the random variable of "static" computation C in the q^{th} incorrect subset. Hence,

$$C \leqslant \sum_{i=0}^{\infty} \sum_{s=0}^{\infty} \sum_{m=1}^{M(s)} \{z_{i,s}(m) + z_{-i,s}(m)\} \tag{72}$$

where M(s) is given by Eq. (1) and the i = 0 term is repeated twice.

We are now prepared to overbound the distribution of computation using a "random code" argument.

B. MOMENTS OF COMPUTATION

Although a lower bound to the distribution of computation $P_R\,[C \geqslant L]$ was found by considering an appropriately chosen subset of the set of events leading to L or more computations, if we are to overbound this distribution, we must consider every event which may lead to L or more computations. We have overbounded the random variable of computation to simplify the analysis and to include each event which might contribute to computation.

The technique which we shall employ to overbound the distribution is to bound the moments of computation and use a generalized form of Chebysheff's Inequality.

Lemma 1. (Chebysheff's Inequality)

If C is a positive random variable, then

$$P_R\,[C \geqslant L] \leqslant \frac{\overline{C^p}}{L^p} \quad p \geqslant 0 \tag{73}$$

where $\overline{C^p}$ is the p^{th} moment of C.

Proof.

$$\overline{C^p} \geqslant \sum_{c \geqslant L} c^p p(c) \geqslant L^p \sum_{c \geqslant L} p(c)$$

where p(c) is the probability that the random variable C assumes valuc c. Q. E. D.

The following two examples indicate the "tightness" that might be expected with Chebysheff's Inequality.

Example 1:— Let C assume values $0, c_o$ with probabilities $1 - a, a$, respectively, then

$$\overline{C^p} = a\,c_o^p \quad \text{and} \quad P_R\,[C \geqslant L] \leqslant a\left(\frac{c_o}{L}\right)^p .$$

For $L = c_o$, the bound is exact.

Example 2:— Let $C \geqslant 1$ be a continuous random variable with density $p(C) = A/(C^{\alpha})$ where $\alpha > 1$ and $A = \alpha - 1$. Then, for $p < \alpha - 1$

$$\overline{C^p} = \frac{A}{\alpha - p - 1} \quad \text{and} \quad P_R\,[C \geqslant L] \leqslant \frac{A}{\alpha - p - 1}\,\frac{1}{L^p} .$$

As p approaches $\alpha - 1$, the moment (hence the bound) becomes indefinitely large. However, the behavior of the tail as a function of L more closely approximates the true tail behavior $1/L^{\alpha - 1}$.

Judging from Example 2 and the fact that the distribution of computation is algebraic, we should expect that the application of Lemma 1 will lead to a bound which degenerates rapidly as the tail behavior of the bound approaches that of the true distribution. This phenomenon will appear in our results.

Moments of computation cannot, as a rule, be computed directly for any arbitrary code. We can, however, compute these moments over the ensemble of all possible tree codes, and deduce that at least onc code has moments lcss than the enscmble average. The ensemble of codes is generated by assigning probabilities to the codes in such a way that each digit (there are ℓ per trce branch) is statistically indcpendent and identically distributed and is assigned with probabilities $\{p_k\}$, that is, channel digit x_k occurs on a branch in a code with probability p_k. Note that we have deliberately chosen the probability assignment used to computc $f(y_j)$, Eq. (64).

As the last topic in this section, we introduce Minkowski's Inequality (see the Appendix for proof).

Lemma 2. (Minkowski's Inequality)

Let $\{w_h\}$, $1 \leqslant h \leqslant H$ be a set of positive random variables. Then

$$\left[\overline{\left(\sum_{h=1}^{H} w_h\right)^p}\right]^{1/p} \leqslant \sum_{h=1}^{H}\left(\overline{w_h^p}\right)^{1/p}, \quad p \geqslant 1 \tag{74}$$

Using this inequality on Eq. (72), the upper bound to the random variablc of computation, we havc as a bound on thc moments the following:

$$\overline{C^p}^{\,1/p} \leqslant \sum_{i=0}^{\infty}\sum_{s=0}^{\infty}\left[\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p}\right]^{1/p} + \sum_{i=0}^{\infty}\sum_{s=0}^{\infty}\left[\overline{\left(\sum_{m=1}^{M(s)} z_{-i,s}(m)\right)^p}\right]^{1/p} \tag{75}$$

where M(s) is defined by Eq. (1) and we use the fact that $z_{i,s}(m) \geqslant 0$.

Evaluating the moments without using Minkowski's Inequality seems to be a practical impossibility because of the number of cross terms which occur. With this inequality we reduce the problem to that of computing moments of computation on incorrect paths at the same length with the same threshold, namely, $\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p}$. If p is an integer, the latter term may be expanded as follows:

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} = \sum_{m_1=1}^{M(s)}, \ldots, \sum_{m_p=1}^{M(s)} \overline{z_{i,s}(m_1), \ldots, z_{i,s}(m_p)} \tag{76}$$

where the terms in the expansion are expectations of a composite characteristic function or probabilities. Since an expansion of this type does not apply to fractional p, we shall limit our attention to integer p.

In following sections, the first term in Eq. (75) will be overbounded. Since the first and second terms differ only in the sign of the index i, we shall find that the bound on the first term can be applied with minor modification to the second term of Eq. (75).

C. PRELIMINARY COUNTING ARGUMENTS

The two terms in Eq. (75) differ in the sign of the index i. This section will deal primarily with the first term, but the discussion here may also be applied directly to the second term.

We are considering the term

$$\sum_{i=0}^{\infty} \sum_{s=0}^{\infty} \left[\overline{\left(\sum_{m=1}^{M(s)} z_{i,\,s}(m) \right)^{p}} \right]^{1/p} . \tag{77}$$

The p^{th} moment term has been expanded in Eq. (76) for integer p, the only case considered.

$$\overline{\left(\sum_{m=1}^{M(s)} z(m) \right)^{p}} = \sum_{m_1=1}^{M(s)} , \ldots , \sum_{m_p=1}^{M(s)} \overline{z(m_1), \ldots , z(m_p)} . \tag{78}$$

The subscripts i, s have been dropped for the remainder of this section.

In Eq. (78), the terms corresponding to $(m_1, m_2, m_3, m_4) = (1, 10, 4, 10)$ and $(4, 1, 10, 1)$ for the case $p = 4$ are equal since $z^n(m) = z(m) = 1$ or 0 and the ordering of characteristic functions in the product does not affect the value of the product. This suggests that many terms in Eq. (78) are equal, since the indices $(m_1, \ldots , m_p)$ are dummy variables. Let us now consider the multiplicity of a particular term.

Assume that the p-tuple of indices $(m_1, m_2, \ldots , m_p)$ contains $t \leqslant p$ distinct elements $\{\theta_1, \theta_2, \ldots , \theta_t\}$. (Each corresponds to a particular incorrect path of s branches.) Since $z(m_1), \ldots , z(m_p) = z(\theta_1), \ldots , z(\theta_t)$, all p-tuples with the set $\{\theta_1, \ldots , \theta_t\}$ as distinct elements have corresponding terms which are equal. Let $W(t, p)$ be the number of such p-tuples. This number is independent of the particular elements in the set of t distinct elements. We bound $W(t, p)$.

$W(t, p)$ may be viewed as the number of ways of placing one ball in each of p distinguishable cells where the balls are of t different colors and each color must appear at least once. The number of such collections of p balls is less than the number of collections one would have if we include the situations where one or more colors do not appear. This larger number is the number of ways of placing t different elements in each of p distinguishable cells, or t^p. Therefore, $W(t, p) \leqslant t^p$.

To underbound $W(t, p)$, we now establish that $W(t, p) \geqslant t\, W(t, p-1)$. Consider $W(t, p-1)$, the number of ways $(p-1)$ balls of t different colors may be placed in $(p-1)$ cells. Consider extending the collection by placing one additional ball with one of the t colors in a p^{th} cell. This new collection contains $t\, W(t, p-1)$ items. It must contain fewer items than does the collection of $W(t, p)$ items because one color appears at least twice and every other color at least once, establishing the desired bound. Iterating this lower bound $(p-t)$ times and observing that $W(t, t) = t!$ we have $W(t, p) \geqslant t^{p-t} t!$ The two bounds are summarized in the following lemma.

Lemma 3.

The number $W(t, p)$ of different p-tuples $(m_1, \ldots , m_p)$ generated from the set of t distinct elements $\{\theta_1, \theta_2, \ldots , \theta_t\}$, each element appearing at least once has the following bounds:

$$\sqrt{2\pi t}\; e^{-t} t^{p} \leqslant W(t, p) \leqslant t^{p} . \tag{79}$$

Proof.

We use the fact that[19]

$$t! \geqslant t^{t} \sqrt{2\pi t}\; e^{-t} .$$

The second and final counting argument anticipates results to be obtained in the next section. First, however, let us rewrite Eq. (78) in terms of W(t, p).

$$\overline{\left(\sum_{m=1}^{M(s)} z(m)\right)^{p}} = \sum_{t=1}^{\min[M(s), p]} W(t, p) \sum_{\substack{\text{all sets of } t \\ \text{distinct elements} \\ \{\Theta_1, \Theta_2, \dots, \Theta_t\}}} \overline{z(\Theta_1), \dots, z(\Theta_t)} \qquad (80)$$

The upper limit on t indicates that the number of elements in a p-tuple $(m_1, m_2, \dots, m_p)$ cannot exceed either p or M(s), the number of values of each index. In constructing the sets of t distinct elements $\{\Theta_1, \Theta_2, \dots, \Theta_t\}$, we draw each Θ_i from a set of M(s) items. They correspond to nodes at penetration s in the incorrect subset and are otherwise labeled as (Θ_a, s), $1 \leqslant a \leqslant t$.

The terms $\overline{z(\Theta_1)\ z(\Theta_2), \dots, z(\Theta_t)}$ in Eq. (80) are probabilities defined on t distinct paths at penetration s in the incorrect subset. These t paths are composed of a number of branches which is less than or equal to ts, since some paths may have branches in common. (See Fig. 15 where the paths involved are checked.) The next section will show that $\overline{z(\Theta_1), \dots, z(\Theta_t)}$ may be bounded in terms of the number of branches on the paths $\{\Theta_1, \dots, \Theta_t\}$. That being the case, any two sets of t different paths with the same number of branches will have the same bound. We now proceed to count the number of sets $\{\Theta_1, \dots, \Theta_t\}$ with an equal number of branches.

The paths $\{\Theta_1, \Theta_2, \dots, \Theta_t\}$ may be visualized by placing a check next to each of these paths (of length s) in the tree. Above every branch on a path ending with a check place a 1 (see Fig. 15). The number of such ones equals the number of branches on these t paths. Let α_r be the number of ones on branches at length r from the reference node and define $\bar{\alpha}$ by $\bar{\alpha} \triangleq (\alpha_1, \dots, \alpha_r, \dots, \alpha_s)$.

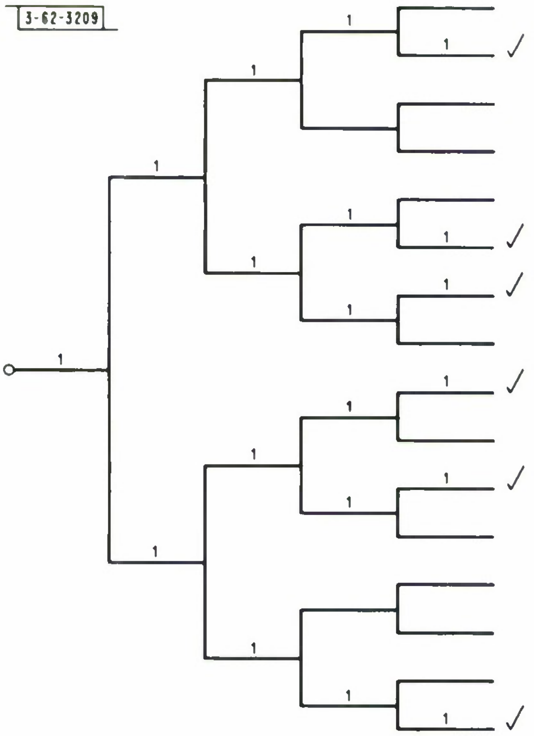


Fig. 15. Topology of tree poths.

In terms of $\bar{\alpha}$, the number of branches on the t paths $\{\theta_1, \dots, \theta_t\}$ equals $\alpha \triangleq \sum_{r=1}^{s} \alpha_r$. Let $N_t(\alpha)$ be the number of sets of t distinct paths $\{\theta_1, \dots, \theta_t\}$ which contain α branches. The following lemma bounds $N_t(\alpha)$.

Lemma 4.

$$N_t(\alpha) \leqslant (t-1)!\ (s+1)^{t-2}\ 2^{\alpha \ell R} \tag{81}$$

where $\alpha = \sum_{r=1}^{s} \alpha_r$; α ranges between $s \leqslant \alpha \leqslant st$.

Proof.

The proof is by construction. We first show that $N_t(\alpha) \leqslant (t-1)!\ s^{t-2} 2^{\alpha \ell R}$ for $s \geqslant 1$. Consider placing the first of the t paths into the incorrect subset of the tree (containing $M(s) \leqslant b^s$ paths). It may assume no more than b^s positions. A second path connecting with the first, but having d_1 separate branches may assume any one of b^{d_1} positions since its point of connection to the first path is fixed by its length d_1. A third path with d_2 branches distinct from the first two may connect to either path and terminate in one of b^{d_2} positions, that is, it can assume no more than $2b^{d_2}$ places. The t^th^ path having d_{t-1} branches distinct from the first $t-1$ paths may be connected to any'one of them and may terminate in any one of $b^{d_{t-1}}$ positions; hence, can be situated in no more than $(t-1)\ b^{d_{t-1}}$ places. Thus, given that the second path has d_1 branches distinct from the first, that the third path has d_2 branches distinct from the first and the second, etc., the number of arrangements of the t paths cannot exceed $(t-1)!\ b^{\alpha}$ where $\alpha = s + d_1 + d_2 + \dots + d_{t-1}$, the number of branches on these paths. All that remains is to determine the number of ways that values may be assigned to $d_1, d_2, \dots, d_{t-2}$. (Note that d_{t-1} is fixed given α and $d_1, \dots, d_{t-2}$.) Since each number d_i represents a portion of a path, $1 \leqslant d_i \leqslant s$, values may be assigned to $d_1, d_2, \dots, d_{t-2}$ in no more than s^{t-2} ways. Hence, the number of arrangements of t paths containing α branches cannot exceed $(t-1)!\ s^{t-2} b^{\alpha}$. Observing that $b = 2^{\ell R}$, we have the desired result for $s \geqslant 1$. We also have $s \leqslant \alpha \leqslant st$ since one path contains s branches and the number of branches on all paths cannot exceed st. Now, when $s = 0$, the bound on $N_t(\alpha)$ is zero. We cannot let this bound be zero since $M(o) = 1$, and we must include the $s = 0$ term. Therefore, replace s by $(s+1)$. Q. E. D.

As mentioned above, the results of the following section show that $z(\theta_i), \dots, z(\theta_t)$ may be overbounded in terms of α. Let this bound be $Q_{i,s}(\alpha)$. We terminate this section by using the counting arguments introduced here to bound Eq. (76).

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant \sum_{t=1}^{\min[M(s),\, p]} W(t,p) \sum_{\alpha=s}^{st} N_t(\alpha)\, Q_{i,s}(\alpha) \tag{82}$$

where $W(t, p)$ and $N_t(\alpha)$ are bounded by Lemmas 3 and 4, respectively. From Lemma 4, the number of values α cannot exceed st.

D. PROBABILITY TERM

Thc purpose of this section is to overbound the probability $\overline{z_{i,s}(\theta_1), \ldots, z_{i,s}(\theta_t)}$ and show that this bound depends on the tree paths $\theta_1, \ldots, \theta_t$ only through α, the number of branehes whieh they eontain. We call this bound $Q_{i,s}(\alpha)$.

Beforc wc proeeed, it is useful to repeat the definition of the random variable $z_{i,s}(\theta_a)$. From Eq. (71) we have

$$z_{i,s}(\theta_a) = \begin{cases} 1 & \text{if } d(\theta_a, s) \geq T_{i-1} \quad \text{and} \quad d\left(\bar{u}_{r_o}, \bar{v}_{r_o}\right) \leq T_{i+1} \\ & \text{for some } r_o \geq 1 \\ 0 & \text{otherwisc} \end{cases} \tag{71}$$

Thc expectation of a product of characteristic funetions sueh as $z_{i,s}(\theta_1) \ldots, z_{i,s}(\theta_t)$ is the joint probability of thc cvents on whieh each eharaeteristic funetion has value one. Thus, we havc that $\overline{z_{i,s}(\theta_1), \ldots, z_{i,s}(\theta_t)}$ is the probability that $d(\theta_1, s) \geq T_{i-1}$, $d(\theta_2, s) \geq T_{i-1}, \ldots, d(\theta_t, s) \geq T_{i-1}$, $d(\bar{u}_{r_o}, \bar{v}_{r_o}) \geq T_{i+1}$ for $r_o = 1$ or 2 or 3 or. . . . This is thc probability of thc union (on r_o) of a set of interseetions. This may be overbounded by the sum of the probabilities of the various intersections. Therefore, we have

$$\overline{z_{i,s}(\theta_1), \ldots, z_{i,s}(\theta_t)} \leq \sum_{r_o=1}^{\infty} P_R \, [d(\theta_a, s) \geq T_{i-1}, \ 1 \leq a \leq t \quad ,$$

$$d(\bar{u}_{r_o}, \bar{v}_{r_o}) \leq T_{i+1}] \quad . \tag{83}$$

Let us reduee Eq. (83) to a more manageable form. We introduee two lemmas to aid in this task. The first is a probabilistic statement and the second is a form of the Chernov Inequality.

Lemma 5.

Let $\{w_h\}$, $1 \leq h \leq H$ be a set of random variables and $\{W_h\}$, $1 \leq h \leq H$ a set of eonstants. Then,

$$P_R \, [w_1 \leq W_1, w_2 \geq W_2, \ldots, W_H \leq W_H]$$

$$= P_R \, [\sigma_h w_h \geq \sigma_h W_h, \ 1 \leq h \leq H] \leq P_R \left[\sum_{h=1}^{H} \sigma_h w_h \geq \sum_{h=1}^{H} \sigma_h W_h \right] \tag{84}$$

where $\sigma_h \geq 0$ for the inequality $w_h \geq W_h$ and $\sigma_h \leq 0$ for the opposite inequality.

Proof.

The equality follows immediately. The inequality follows sinee the seeond event is implied by the first.

Lemma 6.

Let w be a random variable and W some eonstant. Then,

$$P_R \, [w \geq W] \leq 2^{-W} \overline{2^{w}} \quad . \tag{85}$$

Proof.

$$\overline{2^{w}} \geqslant \sum_{w \geqslant W} 2^{w} p(w) \geqslant 2^{W} \sum_{w \geqslant W} p(w) \quad .$$

Q. E. D.

Equation (83) is overbounded with the aid of Lemmas 5 and 6. We use Lemma 5 with $H = t + 1$, $\sigma_a \geqslant 0$ for $1 \leqslant a \leqslant t$ and $\sigma_{t+1} \triangleq \sigma_o \leqslant 0$. Then,

$$\overline{z_{i,s}(\Theta_1), \ldots, z_{i,s}(\Theta_t)} \leqslant 2^{-\left(\sum\limits_{a=1}^{t} \sigma_a\right) T_{i-1} - \sigma_o T_{i+1}}$$

$$\times \sum_{r_o=1}^{\infty} \overline{2^{\sum\limits_{a=1}^{t} \sigma_a d(\Theta_a, \bar{v}) + \sigma_o d\left(\bar{u}_{r_o}, \bar{v}_{r_o}\right)}} \quad . \tag{86}$$

Any optimization now or later of the parameters σ_a, $1 \leqslant a \leqslant t$, is too difficult to be rewarding. Therefore, we let $\sigma_a = 1/(1 + t)$, $1 \leqslant a \leqslant t$, since this selection leads to meaningful results. Recognizing that $T_i = + i\, t_o$, and remembering that $d(\Theta_a, s) = I(\bar{\Theta}_a, \bar{v}_s) - s\ell R$, $d(\bar{u}_{r_o}, \bar{v}_{r_o}) = I(\bar{u}_{r_o}, \bar{v}_{r_o}) - r_o \ell R$ from Eq. (69) and (70), where $\bar{\Theta}_a$ is the set of s branches preceding (Θ_a, s), we further reduce Eq. (86). (It should be remembered that t_o is the separation between criteria whereas t is a variable.)

$$\overline{z_{i,s}(\Theta_1), \ldots, z_{i,s}(\Theta_t)} \leqslant 2^{+t_o\left(\frac{t}{1+t} - \sigma_o\right)}\, 2^{-i t_o\left(\frac{t}{1+t} + \sigma_o\right)}\, 2^{-s\ell R\left(\frac{t}{1+t}\right)}$$

$$\times \sum_{r_o=1}^{\infty} 2^{-\sigma_o r_o \ell R}\, \overline{2^{\frac{1}{1+t} \sum\limits_{a=1}^{t} I(\bar{\Theta}_a, \bar{v}_s) + \sigma_o I\left(\bar{u}_{r_o}, \bar{v}_{r_o}\right)}} \tag{87}$$

where tree paths $\bar{\Theta}_a$, $1 \leqslant a \leqslant t$, are of length s. [Note that Θ_a indicates the node (Θ_a, s), whereas $\bar{\Theta}_a$ is a tree path of s branches preceding (Θ_a, s).]

Now focus attention on the expectation in Eq. (87). The various bounding techniques and choices of parameters to follow are justified by the end result. The following lemma will be needed:

Lemma 7. (Holder's Inequality)

Let $\{w_h\}$, $1 \leqslant h \leqslant H$, be a set of positive random variables and let $\{\nu_h\}$, $1 \leqslant h \leqslant H$, be a set of positive numbers satisfying

$$\sum_{h=1}^{H} \frac{1}{\nu_h} = 1 \quad .$$

Then,

$$\overline{\prod_{h=1}^{H} w_h} \leqslant \prod_{h=1}^{H} \left(\overline{w_h^{\nu_h}}\right)^{1/\nu_h} \quad .$$

Proof. (See the Appendix)

The expectation taken in Eq. (87) is over the ensemble of correct and incorrect sequences and received sequences. Let $\bar{v}$ be a received sequence which includes $\bar{v}_s$ and $\bar{v}_{r_o}$, that is, $\bar{v}$ contains more than r_o or s branches. We may visualize the average in Eq. (87) as consisting of two successive averages, the first taken over the correct and incorrect sequences with the received sequence fixed (indicated with $\overline{}\big|\,\bar{v}$), the second average taken over the received sequence $\bar{v}$ (indicated with $\overline{}\,\bar{v}$). With $\bar{v}$ fixed, correct and incorrect sequences are statistically independent by construction of the "random code" ensemble. This implies that

$$\left.\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a,\, \bar{v}_s)+\sigma_o I\left(\bar{u}_{r_o},\, \bar{v}_{r_o}\right)}}\,\right|_{\bar{v}}$$

$$= \left.\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a,\, \bar{v}_s)}}\,\right|_{\bar{v}} \times \left.\overline{2^{\sigma_o I\left(\bar{u}_{r_o},\, \bar{v}_{r_o}\right)}}\,\right|_{\bar{v}} \tag{88}$$

where the averages are conditioned on $\bar{v}$. The average in Eq. (87) is the average of Eq. (88) over $\bar{v}$. We overbound the average in Eq. (87) using Lemma 7, where the average of that lemma should be considered as an average on $\bar{v}$. We have $H = 2$ and we let $\nu_1 = (1 + t)/t$, $\nu_2 = 1 + t$. Then, we have for the expectation in Eq. (87),

$$\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a,\, \bar{v}_s)+\sigma_o I\left(\bar{u}_{r_o},\, \bar{v}_{r_o}\right)}} \leqslant \left[\overline{\left(\left.\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a,\, \bar{v}_s)}}\,\right|_{\bar{v}}\right)^{(1+t)/t}}^{\,\bar{v}}\right]^{t/(1+t)}$$

$$\times \left[\overline{\left(\left.\overline{2^{\sigma_o I\left(\bar{u}_{r_o},\, \bar{v}_{r_o}\right)}}\,\right|_{\bar{v}}\right)^{1+t}}^{\,\bar{v}}\right]^{1/(1+t)} . \tag{89}$$

Here the average is first carried out over the ensemble of codes with the received sequence fixed and then over the received sequence. Final arguments in this section are concerned with evaluating and bounding these two terms.

From Eq. (69), we have

$$I(\bar{\Theta}_a, \bar{v}_s) = \sum_{r=1}^{s} \sum_{h=1}^{\ell} \log_2 \frac{p\,[v_{rh}/\Theta_{rh}^{a}]}{f(v_{rh})} \tag{90}$$

where v_{rh}, Θ_{rh}^{a} are the h^{th} digits on the r^{th} branch of $\bar{v}_s$, $\bar{\Theta}_a$ respectively, each of s branches. An equivalent statement applies [from Eq. (70)] when $\bar{\Theta}_a$ is replaced by the correct path $\bar{u}_{r_o}$.

Over the ensemble of codes, digits on correct and incorrect paths are statistically independent and identically distributed with probability assignment $\{p_k\}$. We evaluate the second factor in Eq. (89) by observing that $I(\bar{u}_{r_o}, \bar{v}_{r_o})$ is a sum of $r_o\ell$ statistically independent random variables each of which assumes values

$$\left\{\log_2 \frac{p\,[y_j/x_k]}{f(y_j)}\ ,\quad 1 \leqslant j \leqslant J,\quad 1 \leqslant k \leqslant K\right\} \quad .$$

Conditioned upon $\bar{v}$, each of these $r_o\ell$ random variables assumes value

$$\log_2 \left\{\frac{p\,[y_j/x_k]}{f(y_j)}\right\}$$

with probability

$$P_R\,[x_k/y_j] = p_k\,\frac{p\,[y_j/x_k]}{f(y_j)} \tag{91}$$

when the corresponding received digit is y_j. We recall that $f(y_j)$ is the probability of channel output symbol y_j when input symbols are assigned probabilities $\{p_k\}$. For the second factor in Eq. (89), we have

$$\left[\overline{\left(2^{\sigma_o I(\bar{u}_{r_o}, \bar{v}_{r_o})}\Big|_{\bar{v}}\right)^{1+t}}^{\,\bar{v}}\right]^{1/(1+t)} = \left[\sum_{j=1}^{J} f(y_j)\left\{\sum_{k=1}^{K} p_k \left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{1+\sigma_o}\right\}^{1+t}\right]^{r_o\ell/(1+t)} \tag{92}$$

$$= 2^{r_o\ell\mu_t(\sigma_o)} \tag{93}$$

where

$$\mu_t(\sigma_o) \triangleq \frac{1}{1+t}\log_2 \sum_{j=1}^{I} f(y_j)\left\{\sum_{k=1}^{K} p_k\left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{1+\sigma_o}\right\}^{1+t} \quad . \tag{94}$$

Before evaluating the first factor in Eq. (89), let us observe that several of the t paths $\{\Theta_1, \ldots, \Theta_t\}$ at penetration s may have branches in common. We recall that in the previous section we identified branches on the paths $\{\Theta_1, \ldots, \Theta_t\}$ by placing a 1 above each (see Fig. 15). We then defined α_r as the number of branches at length r, that is, the number of 1's on branches at length r. Since $\alpha_r \leqslant t$, a branch at length r may belong to more than one of the t terminal paths. Let δ_n be the number of terminal paths containing the n^{th} of the α_r branches at length r, $1 \leqslant n \leqslant \alpha_r$. Since the total number of terminal paths is t, we have

$$\sum_{n=1}^{\alpha_r} \delta_n = t \quad . \tag{95}$$

(The dependence of δ_n on r is implicit.) Call this n^{th} branch at length r $\underline{\varphi}_r^n$ and let φ_{rh}^n be the h^{th} digit (of ℓ digits) on this branch. Then, in Eq. (89) we have

$$\sum_{a=1}^{t} I(\bar{\Theta}_a, \bar{v}_s) = \sum_{r=1}^{s}\sum_{n=1}^{\alpha_r} \delta_n \sum_{h=1}^{\ell} \log_2 \frac{p\,[v_{rh}/\varphi_{rh}^n]}{f\,[v_{rh}]} \quad . \tag{96}$$

Over the ensemble of codes, the tree digits φ^{n}_{rh} are statistically independent and identically distributed and drawn with probabilities $\{p_k\}$. Since φ^{n}_{rh} is a digit on a branch in the incorrect subset, it is statistically independent of the corresponding transmitted digit and of the corresponding received digit v_{rh}. Therefore, sets of digits $(v_{rh}, \varphi^{1}_{rh}, \ldots, \varphi^{\alpha_r}_{rh})$ are statistically independent of one another as are the digits in each set. Digit v_{rh} assumes value y_j with probability $f(y_j)$, given by Eq. (64). This is the same function $f(y_j)$ appearing in the definitions of the metric. The conditional expectation in the first term in Eq. (98) becomes:

$$\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a, \bar{v}_s)}}\Bigg|_{\bar{v}} = \overline{\prod_{r=1}^{s}\prod_{h=1}^{\ell}\left[\prod_{n=1}^{\alpha_r} 2^{\frac{\delta_n}{1+t}\left\{\log_2 \frac{p\,[v_{rh}/\varphi^{n}_{rh}]}{f(v_{rh})}\right\}}\right]}\Bigg|_{\bar{v}} \tag{97}$$

$$= \prod_{r=1}^{s}\prod_{h=1}^{\ell}\left[\prod_{n=1}^{\alpha_r}\left\{\sum_{k=1}^{K} p_k\left[\frac{p\,[v_{rh}/x_k]}{f(v_{rh})}\right]^{\delta_n/(1+t)}\right\}\right] \tag{98}$$

But the digits v_{rh} are statistically independent; hence, the first term in Eq. (89) becomes, with the aid of Eq. (98), the following:

$$\left[\overline{\left(\overline{2^{\frac{1}{1+t}\sum\limits_{a=1}^{t} I(\bar{\Theta}_a, \bar{v}_s)}}\Bigg|_{\bar{v}}\right)^{(1+t)/t}}^{\,\bar{v}}\right]^{t/(1+t)}$$

$$= \prod_{r=1}^{s}\left[\sum_{j=1}^{J} f(y_j)\prod_{n=1}^{\alpha_r}\left\{\sum_{k=1}^{K} p_k\left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{\delta_n/(1+t)}\right\}^{(1+t)/t}\right]^{\ell t/(1+t)} \tag{99}$$

where we recognize that the random variables in the square brackets of Eq. (98) are statistically dependent.

The above probability is not yet in usable form. As the first of two steps directed at putting it in usable form we use Holder's Inequality (Lemma 7) on Eq. (99) where we identify w_h with

$$\left\{\sum_{k=1}^{K} p_k\left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{\delta_n/(1+t)}\right\}^{(1+t)/t}$$

and we let $\nu_h = t/\delta_n$. We note that

$$\sum_{h=1}^{H}\frac{1}{\nu_h} = \frac{1}{t}\sum_{n=1}^{\alpha_r}\delta_n = 1$$

so that the ν_h satisfy the necessary constraint. Then,

$$\left[\overline{\left(\overline{2^{\frac{1}{1+t}\sum_{a=1}^{t} I(\overline{\Theta}_a, \overline{v}_s)}}\Bigg|_{\overline{v}}\right)^{(1+t)/t}}^{\overline{v}}\right]^{t/(1+t)}$$

$$\leqslant \prod_{r=1}^{s} \prod_{n=1}^{\alpha_r} \left[\sum_{j=1}^{J} f(y_j)\left\{\sum_{k=1}^{K} p_k \left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{\delta_n/(1+t)}\right\}^{(1+t)/\delta_n}\right]^{\delta_n \ell/(1+t)} \tag{100}$$

In the second step we define

$$R_\beta \triangleq -\frac{1}{\beta} \log_2 \sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{1/(1+\beta)}\right)^{1+\beta} \tag{101}$$

and observe that terms in Eq. (100) can be rewritten as follows:

$$\left[\sum_{j=1}^{J}\left(\sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{\delta_n/(1+t)}\right)^{(1+t)/\delta_n}\right]^{\delta_n/(1+t)} = 2^{-\left(1-\frac{\delta_n}{1+t}\right) R_\beta} \tag{102}$$

where

$$\beta = \frac{1+t-\delta_n}{\delta_n} \quad . \tag{103}$$

We note that $\beta \leqslant t$ since $1 \leqslant \delta_n \leqslant \alpha_r \leqslant t$.

Next, we deduce from the following lemma that $+R_\beta \geqslant + R_t$ for $\beta \leqslant t$ so that $-R_\beta \leqslant -R_t$ and Eq. (102) may be overbounded by replacing R_β with R_t.

Lemma 8.

R_β as defined above is a monotone decreasing function of β for $\beta \geqslant 0$.

Proof. (See the Appendix.)

Replacing R_β with R_t in Eq. (102) and inserting this result into the inequality of Eq. (100), we have the following final bound:

$$\left[\overline{\left(\overline{2^{\frac{1}{1+t}\sum_{a=1}^{t} I(\overline{\Theta}_a, \overline{v}_s)}}\Bigg|_{\overline{v}}\right)^{(1+t)/t}}^{\overline{v}}\right]^{t/(1+t)} \leqslant 2^{\frac{s\ell t R_t}{1+t}}\, 2^{-\alpha \ell R_t} \tag{104}$$

where $\alpha = \sum_{r=1}^{s} \alpha_r$ is the number of branches on the set of paths $\{\Theta_1, \ldots, \Theta_t\}$ and we have used Eq. (95). Combining Eqs. (93) and (94) in Eq. (89) we have the following:

$$\overline{2^{\frac{1}{1+t}\sum_{a=1}^{t} I(\overline{\Theta}_a, \overline{v}_s)+\sigma_o I\left(\overline{u}_{r_o}, \overline{v}_{r_o}\right)}} \leqslant 2^{r_o \ell \mu_t(\sigma_o)}\, 2^{\frac{s\ell t R_t}{1+t}}\, 2^{-\alpha \ell R_t} \tag{105}$$

where $\mu_t(\sigma_o)$ and R_t are given by Eqs. (95) and (101), respectively.

Our last step, which is to use Eq. (105) in Eq. (87), is stated formally in the following theorem:

Theorem 4.

The probability $\overline{z_{i,s}(\Theta_1),\ldots,z_{i,s}(\Theta_t)}$ is bounded by the following, where α is the number of branches on the tree paths of length s, $\{\Theta_1,\ldots,\Theta_t\}$:

$$\overline{z_{i,s}(\Theta_1),\ldots,z_{i,s}(\Theta_t)} \leqslant Q_{i,s}(\alpha) \triangleq 2^{+t_o\left(\frac{t}{1+t}-\sigma_o\right)}\, 2^{-it_o\left(\frac{t}{1+t}+\sigma_o\right)}$$

$$\times\, 2^{\frac{s\ell t}{1+t}(R_t-R)}\, 2^{-\alpha\ell R_t}\left(\sum_{r_o=1}^{\infty} 2^{-r_o\ell[\sigma_o R-\mu_t(\sigma_o)]}\right) \qquad (106)$$

where $\sigma_o \leqslant 0$,

$$R_t \triangleq -\frac{1}{t}\log_2 \sum_{j=1}^{J}\left(\sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{1/(1+t)}\right)^{1+t} \qquad [\text{Eq. } (101)]$$

and

$$\mu_t(\sigma_o) \triangleq \frac{1}{1+t}\log_2 \sum_{j=1}^{J} f(y_j)\left(\sum_{k=1}^{K} p_k\left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{1+\sigma_o}\right)^{1+t} \quad . \qquad [\text{Eq. } (94)]$$

(We shall discuss the convergence of the sum in Eq. (106) later.)

This is the result at which this section has been directed. We have obtained a bound on the probability term which depends on the paths $\{\Theta_1,\ldots,\Theta_t\}$ only through α, the number of branches which they contain. An identical proof (which we do not include) shows that the probability term corresponding to negative values of i differs from the bound above only in the sign of i and in the value of σ_o (which we shall call σ_1).

The following section combines the results of this section with the counting arguments of the previous section to obtain the complete bound on the moments of "static" computation.

E. BOUND ON MOMENTS

The purpose of this section is to combine the results of the two previous sections, thereby bounding the moments of computation.

From Eq. (82) we have

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant \sum_{t=1}^{\min[M(s),p]} W(t,p) \sum_{\alpha=s}^{s\ell} N_t(\alpha)\, Q_{i,s}(\alpha) \quad . \qquad [\text{Eq. } (82)]$$

The multiplicities W(t, p) and $N_t(\alpha)$ are bounded by Lemmas 3 and 4 which are repeated here in abbreviated form.

Lemma 3.

$$\sqrt{2\pi t}\; e^{-t} t^p \leqslant W(t,p) \leqslant t^p \quad . \qquad [\text{Eq. } (79)]$$

Lemma 4.

$$N_t(\alpha) \leqslant (t-1)!\ (s+1)^{t-2}\ 2^{\alpha\ell R} \quad . \qquad \text{[Eq. (81)]}$$

The lower bound to the function W(t, p) was introduced in order to establish that the bound in Eq. (82) must grow approximately as t^p. To further overbound Eq. (82) we overbound min [M(s), p] by p. Since $M(s) = (b-1)\,b^{s-1}$ for $s \geqslant 1$, it grows rapidly with s and the minimum will equal p for most values of s. These observations lead to the following bound on Eq. (82):

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant \sum_{t=1}^{p} t^p \sum_{\alpha=s}^{st} (t-1)!\ (s+1)^{t-2}\ 2^{\alpha\ell R}\, Q_{i,s}(\alpha) \quad . \tag{107}$$

We are now prepared to use the results of the preceding section, Theorem 4, namely,

$$Q_{i,s}(\alpha) \leqslant 2^{+t_o\left(\frac{t}{1+t}-\sigma_o\right)}\ 2^{-it_o\left(\frac{t}{1+t}+\sigma_o\right)}\ 2^{\left(\frac{s\ell t}{1+t}\right)(R_t-R)}$$

$$\times\ 2^{-\alpha\ell R_t}\left(\sum_{r_o=1}^{\infty} 2^{-r_o\ell[\sigma_o R-\mu_t(\sigma_o)]}\right) \qquad \text{[Eq. (106)]}$$

This bound and that given above yield

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant \sum_{t=1}^{p} t^p 2^{+t_o\left(\frac{t}{1+t}-\sigma_o\right)}\ 2^{-it_o\left(\frac{t}{1+t}+\sigma_o\right)}\ (t-1)!\ (s+1)^{t-2}$$

$$\times\ 2^{\frac{s\ell t}{1+t}(R_t-R)}\left(\sum_{\alpha=s}^{st} 2^{-\alpha\ell(R_t-R)}\right)\left(\sum_{r_o=1}^{\infty} 2^{-r_o\ell[\sigma_o R-\mu_t(\sigma_o)]}\right) \quad . \tag{108}$$

In the previous section (Lemma 8) we discussed R_t and said that it was monotone decreasing with increasing t. If we choose $R < R_p$, then $R_t \geqslant R_p$ for $t \leqslant p$ and each term in the sum on α is less than 1 and each is overbounded by $2^{-s\ell(R_t-R)}$. (We note that this largest term occurs at $\alpha = s$ which corresponds to the case where the paths $\Theta_1, \ldots, \Theta_t$ are one and the same.) Then,

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant \sum_{t=1}^{p} t^p 2^{+t_o\left(\frac{t}{1+t}-\sigma_o\right)}\ 2^{-it_o\left(\frac{t}{1+t}+\sigma_o\right)}\ t!\,(s+1)^{t-1}$$

$$\times\ 2^{\frac{-s\ell(R_t-R)}{1+t}}\left(\sum_{r=1}^{\infty} 2^{-r\ell[\sigma_o R-\mu_t(\sigma_o)]}\right) \quad . \tag{109}$$

We have yet to discuss whether the sum on r above converges and if so, for what values of σ_o. The semi-invariant moment generating function $\mu_t(\sigma_o)$ is given by

$$\mu_t(\sigma_o) = \log_2\left(\sum_{j=1}^{I} f(y_j)\left\{\sum_{k=1}^{K} p_k\left[\frac{p\,[y_j/x_k]}{f(y_j)}\right]^{1+\sigma_o}\right\}^{1+t}\right)^{1/(1+t)} \quad . \qquad \text{[Eq. (94)]}$$

Using the following lemma, we find that $\mu_t(\sigma_o) \leqslant \mu_p(\sigma_o)$. Thus, if there exists a σ_o sueh that $\sigma_o R - \mu_p(\sigma_o) > 0$ then $\sigma_o R - \mu_t(\sigma_o) > 0$ for $t \leqslant p$.

Lemma 9.

Let w be a positive random variable and $0 < \nu \leqslant n$. Then

$$(\overline{w^\nu})^{1/\nu} \leqslant (\overline{w^n})^{1/n} \quad . \tag{110}$$

Proof. (See the Appendix.)

We must aseertain whether there exists a $\sigma_o < 0$ when $R < R_p$ sueh $\sigma_o R - \mu_p(\sigma_o)$ is positive. If so, the sum on r in Eq. (109) eonverges. The next lemma will aid us in our determination.

Lemma 10.

The funetion $\sigma_o R - \mu_p(\sigma_o)$ where $\mu_p(\sigma_o)$ is given by Eq. (94) is positive for $\sigma' \leqslant \sigma_o \leqslant 0$ where $\mu_p(\sigma')/\sigma' = R$, and $\mu_p(\sigma_o)/\sigma_o$ is monotone inereasing in σ_o.

Proof. (See the Appendix.)

We deduee from the monotonieity of $\mu_p(\sigma_o)/\sigma_o$ that $\sigma' < -p/(1+p)$. Therefore, there exists $\sigma_1 < -p/(1+p)$, $\sigma_o > -\frac{1}{2}$ such that $\sigma_1 R - \mu_p(\sigma_1) > 0$ and $\sigma_o R - \mu_p(\sigma_o) > 0$ when $R < R_p$. We shall need these results soon.

In any further bounding of Eq. (109) we must eonsider the two polarities in i, namely $i \leqslant 0$, $i \geqslant 0$. We bound Eq. (109) over the two ranges of the index i, using the monotonieity in $t/(1+t)$ (up), in R_t (down) and in $\mu_t(\sigma_o)$ (up) with inereasing t.

Theorem 5.

For $i \geqslant 0$, $R < R_p$, and $\sigma_o > \sigma'$

$$\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^p} \leqslant 2^{t_o(1-\sigma_o)}\, 2^{-it_o(\frac{1}{2}+\sigma_o)}\, pp!\,(s+1)^{p-1}\, p^p$$

$$\times\, 2^{\frac{-s\ell(R_p-R)}{1+p}} \left(\sum_{r=1}^{\infty} 2^{-r\ell[\sigma_o R-\mu_p(\sigma_o)]}\right) \quad . \tag{111}$$

For $i \leqslant 0$, replaee σ_o by σ_1 and $2^{-it_o(1/2+\sigma_o)}$ by

$$2^{+it_o(\frac{1}{1+p}+\sigma_1)}$$

Proof.

We note that $\frac{1}{2} \leqslant (t)/(1+t) \leqslant (p)/(1+p)$, using the lower bound for $i \geqslant 0$ and the upper bound for $i \leqslant 0$.

Theorem 5 is now employed to eompute the sum of the two terms in Eq. (75).

Theorem 6.

There exists σ_o, $\sigma_1 \leqslant 0$ such that the following is bounded for $R < R_p$:

$$\sum_{i=0}^{\infty}\sum_{s=1}^{\infty}\left\{\left[\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^{p}}\right]^{1/p}+\left[\overline{\left(\sum_{m=1}^{M(s)} z_{-i,s}(m)\right)^{p}}\right]^{1/p}\right\}$$

$$\leqslant\left(\sum_{s=0}^{\infty}(s+1)^{1-(1/p)}\,2^{\frac{-s\ell(R_p-R)}{(1+p)p}}\right)\left(2^{t_o(1-\sigma_o)}\,pp!\right)^{1/p}p$$

$$\times\left[\left(\sum_{r=1}^{\infty}2^{-r\ell[\sigma_o R-\mu_p(\sigma_o)]}\right)\left(\sum_{i=0}^{\infty}2^{\frac{-it_o(\frac{1}{2}+\sigma_o)}{p}}\right)\right.$$

$$\left.+\left(\sum_{r=1}^{\infty}2^{-r\ell[\sigma_1 R-\mu_p(\sigma_1)]}\right)^{1/p}\left(\sum_{i=0}^{\infty}2^{\frac{+it_o(\frac{p}{1+p}+\sigma_1)}{p}}\right)\right]\quad . \tag{112}$$

Proof.

The discussion following Lemma 10 indicates that for $R < R_p$ there exists $\sigma_o > -\frac{1}{2}$ and $\sigma_1 < -(p)/(1+p)$ such that $\sigma_o R - \mu_p(\sigma_o) > 0$ and $\sigma_1 R - \mu_p(\sigma_1) > 0$. These first two conditions and the last two conditions guarantee convergence of the i and r summations, respectively.

Q. E. D.

We conclude our discussion of the moments with the following theorem which summarizes the results of the last three sections. We recall the bound Eq. (75).

$$\overline{C^p}^{1/p}\leqslant\sum_{i=0}^{\infty}\sum_{s=0}^{\infty}\left\{\left[\overline{\left(\sum_{m=1}^{M(s)} z_{i,s}(m)\right)^{p}}\right]^{1/p}+\left[\overline{\left(\sum_{m=1}^{M(s)} z_{-i,s}(m)\right)^{p}}\right]^{1/p}\right\}\quad . \qquad [\text{Eq. } (75)]$$

Theorem 7.

On the DMC, the pth moment of computation with the Fano Sequential Decoding Algorithm is $\overline{C^p}$, which is considered as an average over the ensemble of tree codes, and is finite for $R < R_p$ where

$$R_p = -\frac{1}{p}\log_2\sum_{j=1}^{J}\left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/(1+p)}\right)^{1+p}\quad . \qquad [\text{Eq. } (101)]$$

A bound to $\overline{C^p}$ is obtained by combining Eq. (75) with Theorem 6.

F. COMPOSITE BOUND ON DISTRIBUTION

Our concern for the moments of computation was motivated earlier by the statement that the moments may be used with a form of Chebysheff's Inequality to bound the distribution of computation. We restate Lemma 1.

Lemma 1.

Let C be a positive random variable with moments $\overline{C^p}$. Then,

$$P_R\ [C \geqslant L] \leqslant \frac{\overline{C^p}}{L^p} \qquad [\text{Eq. (73)}]$$

Since the moments have been averaged ovcr the ensemble of all tree eodes, wc havc a bound on the distribution eonsidered as an average over the ensemble of tree eodes. Indieate this average with $\overline{P_R(C \geqslant L)}$.

It has becn shown abovc that $\overline{C^p}$ is finitc for $R < R_p$. We eannot establish thc exact behavior of $\overline{C^p}$ from our arguments sinee $\overline{C^p}$ has been overbounded. Thereforc, we shall bc contcnt to consider only those momcnts, namely, first, seeond, . . . , p^{th}, such that $R < R_p$. To avoid confusion let k indicate an arbitrary order of moment and define p by $R_{p+1} \leqslant R < R_p$ (note that R_p is monotonc dcereasing in increasing p). Thcrefore, moments of order $k \leqslant p$ convcrge and may be uscd in bounding $\overline{P_R\ [C \geqslant L]}$.

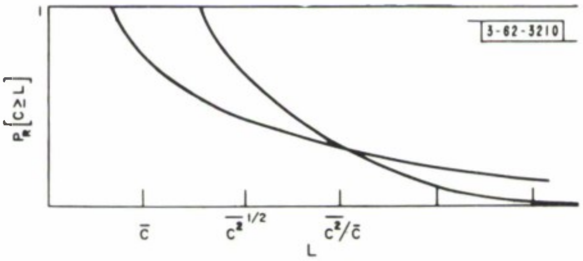


Fig. 16. Bound on distribution.

Given that moments of order $k \leqslant p$ are to be used in bounding the ensemble average of the distribution of eomputation, we ask for that order of moment for which the bound is smallest. If the k^{th} order moment is used and $L \leqslant (\overline{C^k})^{1/k}$, then the bound on the distribution is greater than one, so that one must be used as a bound. Sinee $\overline{C^k}^{1/k}$ inereases with k (Lemma 7), the bound on the distribution must be one for $L < \overline{C}$ and $\overline{C}/L$ for L just greater than $\overline{C}$. This bound is used for valucs L sueh that $\overline{C^2}/L^2$ exceeds $\overline{C}/L$. The point of interseetion of these two curves oeeurs at $L = \overline{C^2}/\overline{C}$ (see Fig. 16). For values of L greater than this value, the seeond-order moment is used until $L = \overline{C^3}/\overline{C^2}$ at whieh point the third-order moment is applied, ete. In general, we use the k^{th} order moment for $\overline{C^k}/\overline{C^{k-1}} \leqslant L \leqslant (\overline{C^{k+1}})/\overline{C^k}$. The eomposite bound is stated below (see Fig. 17).

Theorem 8.

Let C be the random variable of eomputation with moments $\overline{C^k}$ over the ensemble of tree eodes, then, for $k \leqslant p$, where $R_{p+1} \leqslant R < R_p$.

$$\overline{P_R\ [C \geqslant L]} \leqslant \begin{cases} 1 \ , & L \leqslant \overline{C} \\ \overline{C^k/L^k} \ , & (\overline{C^k}/\overline{C^{k-1}}) \leqslant L \leqslant \overline{C^{k+1}}/\overline{C^k} \end{cases} \ . \qquad (113)$$

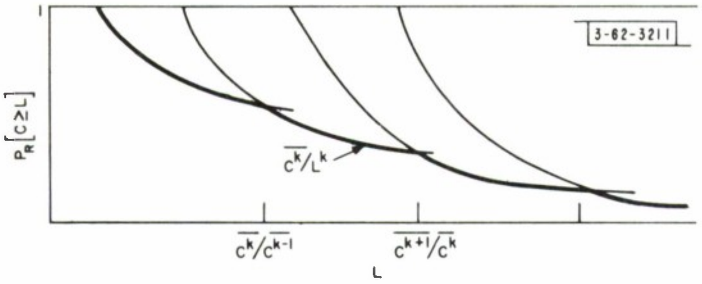


Fig. 17. Composite bound on distribution.

With probability equal to 0.9 a code of rate R chosen at random from the ensemble of codes will have $P_R\,[C \geqslant L] \leqslant 10\, \overline{P_R\,[C \geqslant L]}$. Codes in the ensemble are assigned probabilities in such a way that digits in the code are statistically independent and identically distributed with probabilities $\{p_k\}$.

Proof.

The bound on the average distribution has been discussed above. The second statement follows from Markov's Inequality (a variant of Chebysheff's Inequality), namely, if x is a positive random variable

$$P_R\,[x \leqslant a\bar{x}] = 1 - P_R\,[x \geqslant a\bar{x}] \leqslant 1 - \frac{1}{a}$$

where x is a distribution of computation and a = 10.

The composite bound is the lower envelope of the bounds corresponding to the individual moments. For large L (the distribution parameter) the distribution behaves as L^{-p} where p is the largest order moment which is guaranteed to converge. That is, p is such that $R_{p+1} \leqslant R < R_p$.

CHAPTER V
INTERPRETATION OF RESULTS AND CONCLUSIONS

This report is motivated by a concern for the computational requirements of the Fano Sequential Decoding Algorithm as reflected in the probability of a buffer overflow. This probability plays a central role in the design of the Fano decoder for two reasons:

(a) The probability of an overflow is much larger than the probability of an undetected error (errors without overflow);

(b) When overflows occur a serious break in the decoding process results.

Our particular concern with the overflow event is to determine its sensitivity to the storage capacity of the decoder, to the decoder's speed of operation, and to the signaling rate of the source. We have had to approach these questions indirectly to avoid difficult analytical problems. Our approach has been to consider a random variable of computation known as "static" computation C. We have over- and underbounded the probability distribution of "static" computation, $P_R\ [C \geqslant L]$, and have shown that it behaves as $L^{-\alpha}$, $\alpha > 0$, for large L. The bounds to $P_R\ [C \geqslant L]$ lead to bounds on α.

We shall describe an experiment performed at Lincoln Laboratory and indicate the correlation between this experiment and the analytical bounds on α. This will lead to a conjecture about the true tail behavior of $P_R\ [C \geqslant L]$, i.e., the behavior of this probability for large L. We shall interpret the conjectured exponent α in terms of established bounds on exponents of probabilities of error, these exponents being derived from coding theorems.

In this chapter, we also establish a heuristic connection between the probability of buffer overflow and the distribution of "static" computation $P_R\ [C \geqslant L]$. From this connection we indicate the sensitivities to buffer size, machine speed, and signaling rate which are displayed by the overflow probability. Finally, we introduce and discuss several research problems.

We begin this chapter with a discussion of the tail behavior of $P_R\ [C \geqslant L]$.

A. COMPUTATION EXPONENT

In Chapter III, a lower bound applying to all codes was found for $P_R\ [C \geqslant L]$. A lower bound for codes of fixed composition was also found. We shall be concerned here only with the general lower bound.

In Chapter IV, an overbound to $P_R\ [C \geqslant L]$ was found using the "random code" technique. It was shown that a large fraction of the set of all tree codes have a distribution function $P_R\ [C \geqslant L]$ which is less than some fixed multiple of the ensemble average of $P_R\ [C \geqslant L]$.

It was indicated by Example 2 of Chapter IV that the upper bound on $P_R\ [C \geqslant L]$ of that chapter should be numerically weak. Because of the lower bounding technique described in Chapter III, the same may be said for the lower bound. Example 2 did indicate, however, that the behavior of the upper bound in the distribution parameter L should approximate the true (ensemble average) tail behavior. We are thus motivated to consider the behavior of $P_R\ [C \geqslant L]$ with L for large L. To study this behavior, we introduce a function e(R) called the computation exponent.

$$e(R) \triangleq R \left\{ \lim_{L \to \infty} \left(\frac{-\log_2 P_R\ [C \geqslant L]}{\log_2 L} \right) \right\} \quad . \tag{114}$$

Since $P_R\,[C \geqslant L]$ behaves as $L^{-\alpha}$ for large L, the exponent α is related to the computation exponent e(R) by $\alpha = e(R)/R$. Multiplication by the rate R normalizes α so that e(R) is a bounded function.

We now use the definition of Eq. (114) on Theorems 3 and 8 to obtain upper and lower bounds, respectively, to e(R). We note that e(R) is an implicit function of the code, since $P_R\,[C \geqslant L]$ is a function of the code.

Theorem 9.

On the completely connected DMC, a code cannot be found with a computation exponent exceeding $\bar{e}(R)$ where

$$\bar{e}(R) \triangleq (-\sigma_o)\,(R - I_{min}) \tag{115}$$

and σ_o is the solution to

$$R = \max_k \frac{\gamma_k(\sigma_o)}{\sigma_o} \quad \text{for} \quad -1 \leqslant \sigma_o \leqslant 0 \quad . \qquad \text{[Eq. (60)]}$$

Here, $\gamma_k(\sigma)$ is given by

$$\gamma_k(\sigma) \triangleq \log_2 \sum_{j=1}^{J} p\,[y_j/x_k]^{1+\sigma}\, f(y_j)^{-\sigma} \qquad \text{[Eq. (38)]}$$

$$I_{min} \triangleq \min_{j,k} \log_2 \frac{p\,[y_j/x_k]}{f(y_j)} \qquad \text{[Eq. (17)]}$$

and

$$f(y_j) = \sum_{k=1}^{K} p_k\, p\,[y_j/x_k] \quad . \qquad \text{[Eq. (64)]}$$

Theorem 10.

On the general DMC there exist codcs with computation exponents grcater than or equal to $\underline{e}(R)$ where

$$\underline{e}(R) = p\,R \tag{116}$$

for $R_{p+1} \leqslant R < R_p$, $p = 1, 2, 3, \ldots,$ and

$$R_p \triangleq -\frac{1}{p} \log_2 \sum_{j=1}^{J} \left\{ \sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{1/(1+p)} \right\}^{1+p} \quad . \qquad \text{[Eq. (104)]}$$

The probabilities $\{p_k\}$ are the probabilities assigned to letters in codes in thc "random code" argument. Thcy also appear implicitly in thc definition of thc path metric through thc function $f(y_j)$. Thc path metric on the path terminated by node (m, s, q) of the q^{th} incorrect subset, d(m, s, q), is defined as by

$$d(m, s, q) = \sum_{r=1}^{n} \sum_{h=1}^{\ell} \left\{ \log_2 \frac{p\,[v_{rh}/u_{rh}]}{f(v_{rh})} - R \right\} \quad . \tag{117}$$

Here $\bar{u}_n$, $n = q + s$, represents the given tree path; $\bar{v}_n$ represents the corresponding section of the received sequence; and u_{rh}, v_{rh} are the h^{th} digits on the r^{th} branches of $\bar{u}_n$, $\bar{v}_n$, respectively.

Theorems 9 and 10 delimit the tail behavior of $P_R\ [C \geq L]$ as measured with the computation exponent $e(R)$; $e(R) \leq \bar{e}(R)$ for all codes on the completely connected DMC, and there exist codes on the general DMC such that $e(R) \geq \underline{e}(R)$. We now consider the behavior of the two bounds, $\underline{e}(R)$ and $\bar{e}(R)$, with the signaling rate R.

First consider $\bar{e}(R)$. We wish to show that it is a monotone decreasing function of increasing R. We recall from the discussion of Chapter III that $\gamma_k(\sigma_o)/\sigma_o$ is a monotone increasing function of σ_o. This implies that $R = \max\limits_k\ [\gamma_k(\sigma_o)/\sigma_o]$ is also monotone increasing in σ_o. Moreover, $\gamma_k(\sigma_o)/\sigma_o$ is continuous in σ_o as is $R = \max\limits_k\ [\gamma_k(\sigma_o)/\sigma_o]$. If we can show that $\bar{e}(R) = (-\sigma_o)\ (R - I_{min})$ is monotone decreasing in increasing σ_o, we will have established that $\bar{e}(R)$ is a continuous decreasing function of R. The monotonicity of $(-\sigma_o)\ (R - I_{min})$ is established by considering its derivative in σ_o. The derivative is taken at a value of σ_o which is not a transition point of $\max\limits_k\ [\gamma_k(\sigma_o)/\sigma_o]$, that is, a point at which the index which achieves the maximum is changing from $k = k_1$ to $k = k_2$.

$$\frac{d}{d\sigma_o}\ (-\sigma_o)\ (R - I_{min}) = \frac{d}{d\sigma_o}\left[-\gamma_{k_1}(\sigma_o) + \sigma_o I_{min}\right]$$

$$= -\left[\gamma'_{k_1}(\sigma_o) - I_{min}\right] \tag{118}$$

where

$$\gamma'_{k_1} = \frac{\sum\limits_{j=1}^{J} p\left[y_j/x_{k_1}\right]^{1+\sigma_o} f(y_j)^{-\sigma_o} \log_2 \dfrac{p\left[y_j/x_{k_1}\right]}{f(y_j)}}{\sum\limits_{j=1}^{J} p\left[y_j/x_{k_1}\right]^{1+\sigma_o} f(y_j)^{-\sigma_o}} \tag{119}$$

We may underbound each of the $\log_2 \{p\ [y_j/x_{k_1}]/f(y_j)\}$, appearing in Eq. (119), by the smallest such term. By definition, this must exceed I_{min}. Therefore, $\gamma'_{k_1}(\sigma_o) \geq I_{min}$ and $(-\sigma_o)\ (R - I_{min})$ has a negative first derivative at values of σ_o which are not transition points. Since $(-\sigma_o)$ $(R - I_{min})$ is continuous in σ_o, we have that $\bar{e}(R)$ is continuous and monotone decreasing in R. At $\sigma_o = -1$, $R = 0$ and $\bar{e}(R) = -I_{min} \geq 0$. At $\sigma_o = 0$, $R = \lim\limits_{\sigma_o \to 0}\ \max\limits_k\ [\gamma_k(\sigma_o)/\sigma_o] = \max\limits_k\ \gamma'_k(0)$ [since $\gamma_k(0) = 0$] and $\bar{e}(R) = 0$. These results are summarized in the following lemma.

Lemma 12.

The computation exponent upper bound $\bar{e}(R)$ is continuous and monotone decreasing in increasing R. It decreases from $\bar{e}(R) = -I_{min}$ at $R = 0$ to $\bar{e}(R) = 0$ at $R = \max\limits_k\ \gamma'_k(0)$. The computation exponent bound $\bar{e}(R)$ is sketched in Fig. 18 for a typical channel and a typical probability assignment $\{p_k\}$.

One may show that the rate at which $\bar{e}(R) = 0$, namely $\max\limits_k\ \gamma'_k(0)$, may exceed channel capacity. On the contrary, if the assignment $\{p_k\}$ that achieves channel capacity C_o is used then

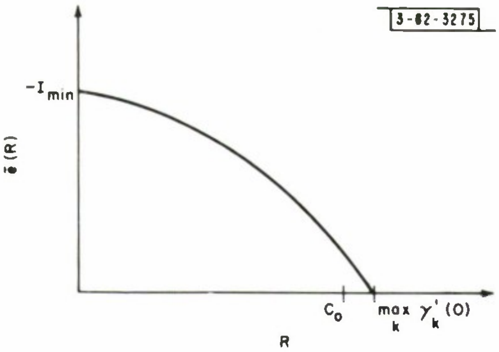


Fig. 18. Computation exponent upper bound $\bar{e}(R)$.

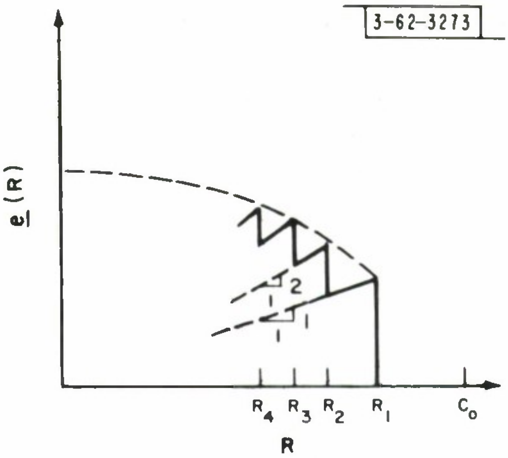


Fig. 19. Computation exponent lower bound $\underline{e}(R)$.

$\max_k \gamma_k'(0) = C_o$. We recall that channel capacity C_o is defined as the maximum mutual information between channel inputs and outputs. Let $I(x,y)$ be the mutual information between channel inputs and outputs; then,

$$C_o \triangleq \max_{\{p_k\}} I(x,y) = \max_{\{p_k\}} \sum_{j=1}^{J} \sum_{k=1}^{K} p_k\, p\,[y_j/x_k] \log_2 \frac{p\,[y_j/x_k]}{f(y_j)} \quad . \qquad (120)$$

It has been shown[20] that the $\{p_k\}$ which maximizes $I(x,y)$ is such that

$$\sum_{j=1}^{J} p\,[y_j/x_k] \log_2 \frac{p\,[y_j/x_k]}{f(y_j)} \leq C_o \quad ; \quad k = 1, 2, \ldots, K \qquad (121)$$

with equality when $p_k \neq 0$. Therefore, if this set $\{p_k\}$ is used in the definition of $f(y_j)$, that is, in the definition of the path metric, then $\max_k \gamma_k'(0) = C_o$ and the rate at which $\bar{e}(R) = 0$ is channel capacity.

We shall now consider the behavior of $\underline{e}(R)$ with R. As given by Theorem 10, $\underline{e}(R) = pR$ for $R_{p+1} \leq R < R_p$, $p = 1, 2, \ldots$. Fix p. Then, for $R_{p+1} \leq R < R_p$, $\underline{e}(R)$ increases with R on a line of slope p passing through the origin. The full curve $\underline{e}(R)$ is sketched in Fig. 19. For R arbitrarily close to, but less than R_p, $\underline{e}(R) = pR_p$. We now show that the points pR_p form an increasing sequence for increasing p, whereas the R_p form a decreasing sequence. This will establish that the sketch of Fig. 19 is accurate.

From Lemma 8, R_β, $\beta \geq 0$, is monotone decreasing in increasing β. We show that pR_p is monotone increasing in p by showing that 2^{-pR_p} is monotone decreasing in p for a fixed set of $\{p_k\}$.

$$2^{-pR_p} = \sum_{j=1}^{J} \left\{ \sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{1/(1+p)} \right\}^{1+p} \qquad (122)$$

Lemma 9 is sufficient to establish the monotonicity of 2^{-pR_p}. We repeat this lemma here.

Lemma 9.

Let w be a positive random variable and let $0 < \nu \leqslant \eta$. Then,

$$(\overline{w^{\nu}})^{1/\nu} \leqslant (\overline{w^{\eta}})^{1/\eta}$$

Therefore, if we apply this lemma to the sum over k for each j in Eq. (122), we find that increasing p decreases 2^{-pR_p} or increases pR_p.

We now show that on the completely connected DMC, pR_p has a well-defined, nonzero limit as $p \to \infty$. For large p,

$$p\,[y_j/x_k]^{1/(1+p)} \triangleq \exp\left\{\frac{1}{1+p}\ \ln p\,[y_j/x_k]\right\} \simeq 1 + \frac{1}{1+p}\ \ln p\,[y_j/x_k] \qquad (123)$$

and

$$\left\{\sum_{k=1}^{K} p_k\, p\,[y_j/x_k]^{1/(1+p)}\right\}^{1+p} \simeq \exp\left[(1+p)\ \ln\left\{1 + \frac{1}{1+p}\sum_{k=1}^{K} p_k\ \ln p\,[y_j/x_k]\right\}\right]$$

$$\simeq \exp\left\{\sum_{k=1}^{K} p_k\ \ln p\,[y_j/x_k]\right\} \quad . \qquad (124)$$

Therefore, on the completely connected DMC, as p becomes indefinitely large, pR_p approaches

$$\log_2 \sum_{j=1}^{J} 2^{\sum_{k=1}^{K} p_k \log_2 p[y_j/x_k]} \quad .$$

This implies that $R_p = pR_p/p$ approaches zero on the completely connected DMC. When the channel is not completely connected, the limit of pR_p as $p \to \infty$ may be infinite. This implies that $R_p \to C_o^+ > 0$. These results are summarized in the following lemma.

Lemma 13.

The computation exponent lower bound $\underline{e}(R)$ is a set of straight lines of increasing slope, $\underline{e}(R) = pR$ for $R_{p+1} \leqslant R < R_p$, $p = 1, 2, 3, \ldots$. On the completely connected DMC the points pR_p increase with decreasing R_p to the following limits

$$\lim_{p\to\infty} R_p = 0 \quad , \quad \lim_{p\to\infty} pR_p = \log_2 \sum_{j=1}^{J} 2^{\sum_{k=1}^{K} p_k \log_2 p[y_j/x_k]} \quad .$$

When the channel is not completely connected $\lim\limits_{p\to\infty} R_p = C_o^+$ where C_o^+ may be strictly positive, $C_o^+ > 0$.

The largest rate for which $\underline{e}(R)$ is nonzero is R_1. For $R \geqslant R_1$, $\underline{e}(R)$ is zero. It will be obvious from a later discussion that $R_1 \leqslant C_o$, channel capacity.

As an example of the computation exponent bounds, we show in Fig. 20 the two exponents $\overline{e}(R)$ and $\underline{e}(R)$ for the binary symmetric channel (BSC) with transition probability $p_o = 0.01$. We

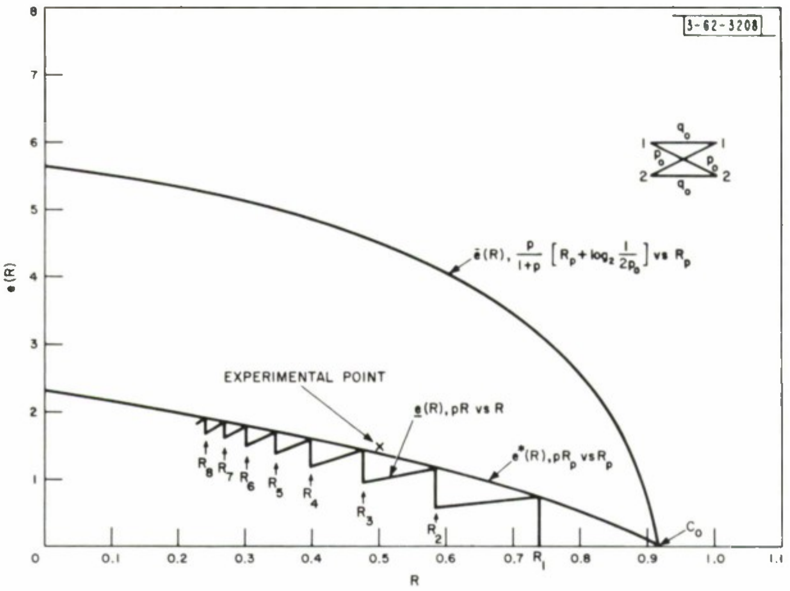


Fig. 20. Bounds on the computation exponent for BSC with $p_o = 0.01$.

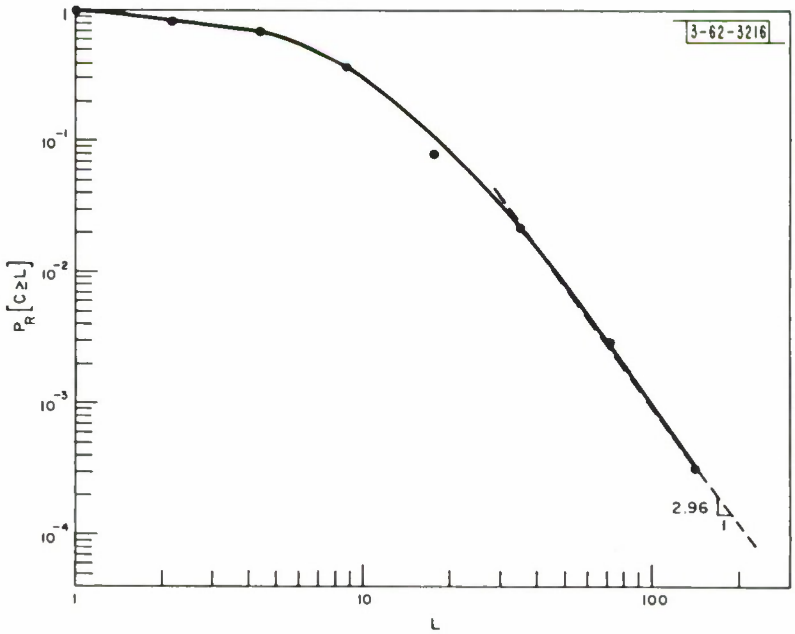


Fig. 21. Empirical distribution of computation.

select $p_k = \frac{1}{2}$, $k = 1, 2$. Since this assignment achieves channel capacity, $\bar{e}(R) = 0$ at $R = C_o$. For this channel and the given assignment $\{p_k\}$, we have $\bar{e}(R) = (p/1 + p)\,[R_p + \log_2 (1/2p_o)]$, where $R = R_p$ and p assumes all values greater than zero (not just the integers). At $R = 0$, $\bar{e}(R) = -I_{min} = \log_2 (1/2p_o)$.

In the next section we correlate the analytical results with an experiment.

B. AN EXPERIMENTAL RESULT

A computer simulation[13] of the Fano algorithm was run recently at Lincoln Laboratory under the direction of K. L. Jordan who has made data from this experiment available to the author. These data represent slightly more than one million decoded digits on the BSC with $p_o = 0.01$ and have been used to compute an experimental distribution of computation (see Fig. 21). The computation variable measured in this simulation will be discussed shortly. It suffices to say that it differs somewhat from "static" computation.

In the experiment, a convolutional tree code of the type described in Chapter II with $b = 2$ was used. In the generator $\underline{g} = (\underline{g}_1, \underline{g}_2, \ldots, \underline{g}_S)$, $S = 60$; $\underline{g}_1$ was chosen to maximize the Hamming distance between the two tree branches at the first node of the tree. Given $\underline{g}_1$, $\underline{g}_2$ is chosen to maximize the minimum Hamming distance between the four codewords of two branches. Several other subgenerators were chosen in this way. The remainder were chosen at random. The BSC was simulated with a random number generator and as the decoder operated, it was assumed to have an infinite buffer.

The computation variable recorded by the computer is best defined with the aid of two imaginary pointers. We may visualize a pointer "extreme" below the tree code indicating the furthest penetration into the tree made by the decoder. Another pointer, "search," below the tree indicates the depth of the node presently being examined by the decoder. The search pointer either lies on or behind the extreme pointer. Every time the two pointers move ahead together in the tree, the computer program records one computation. If a search is required, the extreme pointer remains fixed and the program records the number of operations required before the search pointer returns to the extreme pointer and the two move ahead. The data from the simulation are reduced and the computer program prints out the number of times the computation exceeds 2^k for $k = 0, 1, 2, \ldots$. In the particular run used by the author the signaling rate R was $\frac{1}{2}$ bit per channel use. The largest number of computations in this run was less than 256 and greater than 128 and it was observed that the search pointer never drifted back more than 45 branches from the extreme pointer.

Although the computation recorded by the program is not "static" computation, we shall argue later that it is a small multiple of "static" computation. Since this multiple does not affect the tail behavior of the experimental distribution, we are justified in computing the computation exponent for the experimental distribution and comparing this exponent to the bounds of Fig. 20. The experimental point is shown in Fig. 20. Other computer runs at rates $R = \frac{1}{3}, \frac{1}{4}$ were recorded but large computations were so infrequent that the data were not considered reliable and were not used.

In the next section, we conjecture about the true value of the computation exponent.

C. A CONJECTURE

We are led to conjecture a form for the "true" computation exponent by consideration of the experimental result of the last section and the derivation of the "random code" bound on the distribution of "static" computation. In the discussion of this bound in Chapter IV, we limited attention to integral moments of eomputation for analytieal reasons. As a result of this limitation, $\underline{e}(R)$ has the shape of Fig. 19. We now suggest that the true "random code" computation exponent has the form $e^*(R) = pR_p$ when $R = R_p$ for all $p \geqslant 0$ (not just integer p). We suggest that this is an exponent which may be achieved, that is, that codes can be found with this exponent. (This is partially substantiated by the experimental point discussed in the last section. The conjectured "random code" eomputation exponent and this point differ by only 5 percent at $R = \frac{1}{2}$ for the BSC example.) Finally, we suggest that $e^*(R)$ cannot be exceeded, that is, that no code exists with a computation exponent which exceeds $e^*(R)$. These suggestions are summarized below.

Conjecture

The computation exponent $e^*(R)$,

$$e^*(R) = pR_p \quad , \quad R = R_p \quad \text{for} \quad p \geqslant 0 \tag{125}$$

cannot be exceeded by any code used with the path metric of Eq. (114) and codes exist which achieve this computation exponent.

The eonjectured exponent $e^*(R)$ is a monotone decreasing function of R. This may be deduced from the earlier diseussion of the exponent $\underline{e}(R)$. The value of $e^*(R)$ at $R = 0$ is identical with the value of $\underline{e}(R)$ at $R = 0$. The exponent $e^*(R)$ is zero for $p = 0$ or $R = I(x,y)$ where $I(x,y)$ is given by Eq. (120).

The conjectured exponent of this section is interpreted in the following section in terms of "list deeoding" exponents and the "sphere-packing" exponent.

D. INTERPRETATION OF COMPUTATION EXPONENT

The eonjectured computation exponent $e^*(R)$ has a simple interpretation in terms of the "list decoding exponent," that is, the exponent of the "random code" bound on the probability of error with "list decoding."[21-23]

"List decoding" is similar to maximum a posteriori decoding. We assume that one of $M \triangleq 2^{nR}$ equally likely codewords is transmitted over the DMC. Here n is the code block length in channel symbols and R is the signaling rate. At the receiving terminal, the decoder makes a list of the k a posteriori most probable codcwords given the received channel sequence. If the transmitted codeword is not in this list of k codewords, an error is said to have occurred. With "list decoding" the probability of error is reduced from the probability of error with maximum a posteriori decoding, k = 1, by accepting some ambiguity in the transmitted message.

The probability of error with list decoding has been overbounded using a "random code" argument. The probability of error is averaged over the ensemble of codes by assigning to each code a probability, computed as if each letter in the code were chosen independently with the assignment $\{p_k\}$, the assignment of Chapter IV. The ensemble average of the probability of error with list size k, $P_k(\epsilon)$, $k = 1, 2, 3, \ldots$, has the following bound

$$P_k(\epsilon) \leqslant 2^{-nE_k(R)} \tag{126}$$

where

$$E_k(R) = \max_{0 \leqslant p \leqslant k} [pR_p - pR] \quad . \tag{127}$$

The exponent $E_k(R)$ is the upper envelope of the straight lines $pR_p - pR$ for all $0 \leqslant p \leqslant k$ (see Fig. 22). At $R = I(x,y)$, $E_k(R) = 0$. For $R \leqslant R_k^*$, the point of tangency of the straight line of slope $-k$ to the curve $E_\infty(R) \triangleq \lim_{k\to\infty} E_k(R)$, the exponent $E_k(R)$ increases along a straight line of slope $-k$ to kR_k. The limiting exponent $E_\infty(R)$, as well as $E_k(R)$, depends on the probability assignment $\{p_k\}$. If $E_\infty(R)$ is maximized on $\{p_k\}$, one finds that the resulting exponent equals the "sphere-packing" exponent.^{24} This latter exponent is an exponent† on a lower bound to the probability of error which applies to every block decoding procedure, list decoding or otherwise, and as such the "sphere-packing" exponent represents the largest possible exponent on the probability of error with any block decoding procedure. It is a fundamental bound on exponents to the probability of error.

We now return to the conjectured computation exponent $e^*(R)$. A simple construction on $E_\infty(R)$ yields $e^*(R)$ (see Fig. 23). From R a straight line tangent to $E_\infty(R)$ is drawn; $e^*(R)$ is the height of the intersection with the exponent axis. This straight line has equation $pR_p - pR$ for some p by definition of $E_\infty(R)$, where p is the magnitude of the slope of the tangent line.

Although the conjectured computation exponent [which equals $\underline{c}(R)$ for $R = R_p$, $p = 1, 2, \ldots$] has an interpretation in terms of the "list decoding exponent" and the "sphere-packing" exponent, there is no obvious connection between them. Since the latter two exponents are fundamental in a sense, the fact that the conjectured exponent is interpreted from them suggests that this exponent may also be fundamental. Unfortunately, there is no other evidence to suggest that this is the case.

E. OVERFLOW QUESTION

In this section, we establish a heuristic connection between the probability distribution of "static" computation, which we have studied extensively, and the probability of buffer overflow. Our discussion will indicate the sensitivity of the overflow probability to signaling rate R to machine speed, to buffer size and to the number of digits decoded before overflow. We begin by summarizing the discussion of Chapter II on the overflow event.

We assume that the Fano decoder operates with the buffer shown in Fig. 24. Branches arrive from the channel and are inserted at the left-hand end of the buffer. They move through the buffer at the rate at which they arrive and are released when they reach the right-hand side of the buffer. Below each branch, space is provided to record tentative decisions on the source digits. This portion of the buffer is empty to the left of the pointer "search."

As the decoder proceeds, it inserts or erases tentative source decisions recorded below the tree branches. These insertions or erasures occur at the search pointer because this pointer indicates the received tree branch presently being examined by the machine. The pointer "extreme" indicates the latest received tree branch examined to date. Branches to the left of this pointer have never been compared to branches in the tree code.

The search and extreme pointers hover near the left-hand side of the buffer when the decoder has little trouble decoding. Occasionally, however, an interval of high channel noise forces a large amount of computation and the two pointers drift to the far right end of the buffer. When this happens, there is a high probability that an erroneous digit will be released into the safety zone. Since the decoder is unable to change digits in the safety zone (the corresponding received

† The exponent is defined as $\lim_{n\to\infty} [-\log_2 P(\epsilon)]/n$.

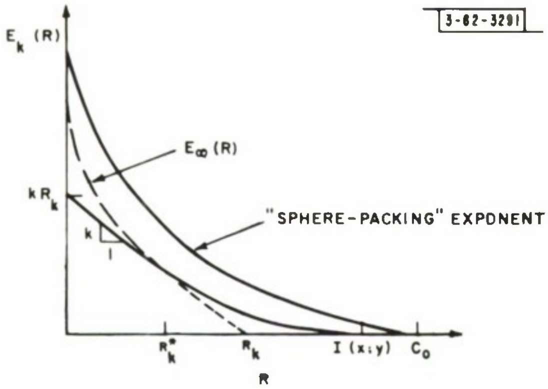
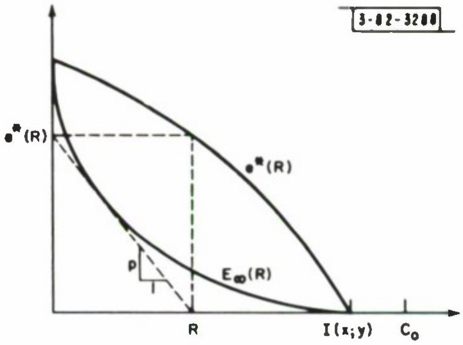


Fig. 22. List decoding exponent.

Fig. 23. Construction for e*(R).



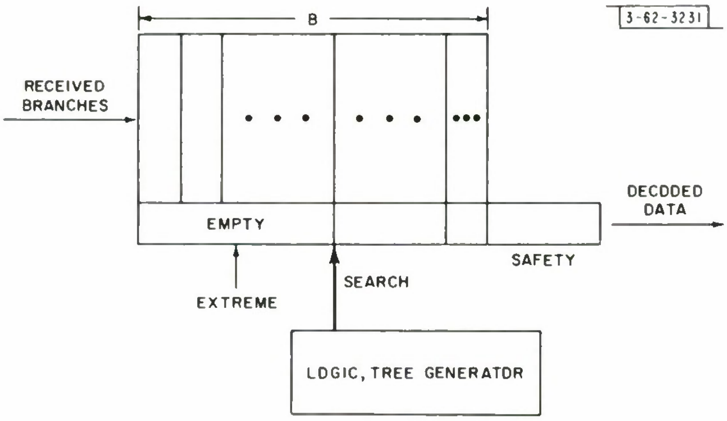


Fig. 24. Buffer.

branches have been discarded), the decoder is forced to consider extending on incorrect paths. This is very difficult, so that thereafter both pointers tend to hover near the far end of the buffer, releasing erroneous digits. Although overflow can be detected, it is a serious disturbance and must be combated either with the use of a feedback channel or periodic resynchronization or by some other means. We will attempt to estimate the sensitivity of the overflow probability to the system parameters.

Now that we understand the meaning of overflow, we return to a consideration of "static" computation. Our intention is to lay the groundwork for a discussion of $P_{BF}(N)$, the probability of a buffer overflow on or before the time at which the N^{th} source decision enters the safety zone.

Consider the q^{th} node on the correct path (1, 0, q). "Static" computation associated with q^{th} correct node is defined as the computation eventually performed with the Fano algorithm on nodes of the q^{th} incorrect subset when the correct message is ultimately decoded. We now argue that whatever computation is performed in this incorrect subset is performed on nodes which are close to the reference node (1, 0, q) and that almost all of these computations are performed together in time rather than a substantial fraction now and a comparable fraction later. We are in effect going to argue that "static" computation is very closely related to "dynamic" computation. The argument is as follows:

(1) For a properly chosen code and for a reasonable range of signaling rates, $R < R_1$, computation in an incorrect subset is due almost completely to an interval of high channel noise and a concomitant dip in the correct path. We argue that this is true by noting that if the correct path does not dip, the decoder will never be searching far from the correct path.

(2) Let W be the width of a dip in the correct path (the separation between points A and B in Fig. 25). Let the magnitude of the dip remain fixed. Then it can be shown that a dip of width W occurs with a probability which decreases exponentially fast in W. Therefore, this width will typically be small.

(3) If the q^{th} correct node (1, 0, q) is in the region of a dip in the correct path (see Fig. 25), then paths in the associated incorrect subset may be above the minimum of the dip over the region A to B of Fig. 25, but beyond B they will typically fall rapidly below the dip minimum never to be extended.

(4) It is conceivable that a dip far ahead of a particular correct node will force a return to the incorrect subset associated with this node. The probability of such an event is very small as is seen from the following observations: Typically, the correct path will rise from a particular

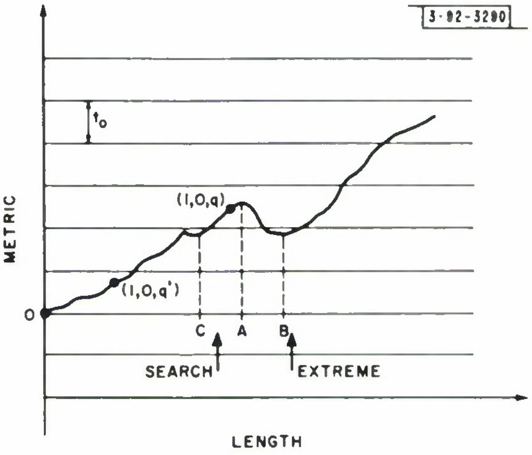


Fig. 25. Typical correct path trajectory.

correct node [see (1, 0, q') of Fig. 25]. If a later dip in the correct path is to force a return to node (1, 0, q'), this dip will have to equal or exceed the rise which previously occurred in the correct path. If such a dip occurs far in the future, it will typically be very large in magnitude. Such an event is very unlikely. It occurs with a probability which decreases exponentially in the magnitude of the dip.[25]

(5) Thus, if computation is required in the qth incorrect subset, with high probability it will be due to a dip in the correct path which is close to the qth correct node. Since the width of the dip will typically be small, all the computation performed in the qth incorrect subset is usually performed on nodes close to the qth correct node. The behavior of the probabilities mentioned in (2) and (4) can be established with a "random code" argument.

Statement 5 summarizes the argument which suggests that "static" computation is related to "dynamic" computation. We note that the "static" computations in the adjacent incorrect subsets, which are located within the region of a correct path dip (C to B in Fig. 25) will be comparable so that the total "dynamic" computation due to the dip will be a small multiple, say N_{avg}, of the "static" computation in one incorrect subset. We also note the pointers "search" and "extreme" indicated in the buffer description may also be applied to the path trajectories of Fig. 25. As a result of a correct path dip, the extreme pointer will move out to point B and will typically remain there until the running threshold has been reduced sufficiently to pass the correct path. It is this argument which justifies our comparing the computation exponent bounds to the data taken from the Lincoln Laboratory simulation. We may also observe from the discussion of Chapter III that the computation increases exponentially with the width of the correct path dip so that for a dip which causes a large computation, the extreme pointer of Fig. 24 will drift back by an amount x while the extreme and search pointers will have a separation proportional to log x. We are now prepared to discuss the overflow probability.

The buffer overflow probability $P_{BF}(N)$ is defined as the probability that overflow occurs on or before the time at which the N^{th} source decision reaches the safety zone. It certainly exceeds $P_{BF}(1)$, that is,

$$P_{BF}(N) \geqslant P_{BF}(1) \quad . \tag{128}$$

First, we shall consider $P_{BF}(1)$ in order to bring out the dependence of $P_{BF}(N)$ on signaling rate R, machine speed, and buffer size.

$P_{BF}(1)$ is the probability that the buffer overflows on or before the time at which the first source decision reaches the safety zone. Since the buffer is empty before the first received branch enters the buffer, overflow can occur if computation in the first incorrect subset and adjacent subsets is sufficient to force the search pointer from the left- to the right-hand side of the buffer. Large computation in these subsets (let there be N_{avg} of them) is due to a local dip in the correct path so that if the total "static" computation over these N_{avg} incorrect subsets exceeds L_o, where L_o is the number of computations needed to force the search pointer to the far end of the buffer, then overflow occurs. If T_{ch} is the time between branch arrivals and B is the number of branches which may be stored in the buffer, then it takes BT_{ch} seconds to fill the buffer. We neglect the distance between the search and extreme pointers and assume that each computation requires γ_m seconds. Then if $L_o = BT_{ch}/\gamma_m$ or more computations are required in the first N_{avg} incorrect subsets, then overflow will result. If the computation in these subsets is comparable, and if the "static" computation in each one of them exceeds $BT_{ch}/N_{avg}\gamma_m$, overflow occurs. Therefore,

$$P_{BF}(N) \geq P_{BF}(1) \sim P_R \left[C \geq L = \frac{BT_{ch}}{N_{avg}\gamma_m} \right] \quad . \tag{129}$$

We may deduce from the fact that $P_R\ [C \geq L]$ behaves as $L^{[-e(R)]/R}$, for large L, where e(R) is the computation exponent, that $P_{BF}(N)$ is relatively insensitive to a change in B, the storage capacity of the buffer, or to a change in γ_m, the time for one machine computation. $P_{BF}(N)$ is very sensitive to signaling rate, however, because the exponent $[e(R)]/R$ increases rapidly with a decrease in rate. These are the sensitivities mentioned in Chapter II. Let us now consider the sensitivity of $P_{BF}(N)$ to N.

It should be clear that $P_{BF}(N)$ will increase rapidly to one with N, the number of source decisions released into the safety zone, if the average number of decoding operations required by the Fano algorithm exceeds the number of computations per second which the decoder can perform. We find from inspection of the conjectured computation exponent that the average computation required by the algorithm is very large if $R \geq R_1$. Therefore, $P_{BF}(N)$ must grow rapidly to one with N for $R \geq R_1$. This then is an upper limit to the rate at which the Fano algorithm may operate with infrequent overflows. It has been shown that the average computation is small if $R \leq 0.9\ R_1$, being several computations per decoded digit. Thus, if the machine speed is such that several times this number of computations per second can be performed, then we do not expect $P_{BF}(N)$ to grow rapidly with N. In fact, one may reasonably argue that decreasing the signaling rate rapidly decreases the probability of frequent intervals of large "dynamic" computation, and this implies that with a reduction in signaling rate the machine decodes easily and both the search and extreme pointers hover near the left-hand end of the buffer. If large computations are infrequent, we expect only one burst of computation at a time, which is to say, that bursts will be statistically independent. $P_{BF}(N)$ then is proportional to N and $P_{BF}(1)$, that is,

$$P_{BF}(N) \simeq N P_{BF}(1) \tag{130}$$

when $R \leq 0.9\ R_1$, $P_{BF}(1)$ is small, and the machine speed exceeds by several times the speed required to handle the average computation.

While the statements of this section are strictly heuristic, there is good reason to believe Eq. (129) because of the experimental result cited above. The statement of Eq. (130) is less secure than that of Eq. (129). At best, it may serve as a guideline.

This completes the discussion of overflow probability.

F. SOME RESEARCH PROBLEMS

We conclude this chapter with a discussion of some problems suggested by the results of this report. We shall discuss these suggested problems in inverse order of importance.

The distribution of "static" computation and the probability of buffer overlow were loosely connected in the previous section. It is unfortunate that the connection had to be heuristic. Perhaps a more direct connection is possible.

If a direct, nonheuristic, approach to the probability of buffer overflow cannot be found, then the heuristic approach of the last section should be improved by improving the bounds on the distribution of "static" computation. In particular, there is reason to believe that a stronger lower bound argument than that presented in Chapter III may be found and that such a bound would not require the assumption that the DMC is completely connected.

A more important problem than the two suggested, concerns the choice of a path metric. The metric assumed for this report, Eq. (117), requires exact knowledge of the channel transition probabilities. There are several reasons for not using a metric of this type.

(1) It may be too difficult to measure the channel transition probabilities;

(2) The channel may be time varying so that a metric for the poorest channel state may be necessary;

(3) The channel transition probabilities may be known but they may be either so large in number or sufficiently difficult to compute in the decoder that some other metric is desirable.

Thus, there is a need to consider the performance of the Fano Sequential Decoding algorithm with a variety of metrics. If we choose to measure the performance of the algorithm with the computation exponent, an analytical treatment of the various metrics may be possible using the technique of Chapter III. It is not expected that the "random code" argument will carry through for many different metrics. It is more reasonable to expect, however, that a fruitful study of the effect of a change in metric on the Fano algorithm will be achieved through simulation. A preliminary study of this type has been completed at Lincoln Laboratory.[13] The behavior of the Fano algorithm appears to be insensitive to a variation in metric.

We come now to the most important problem area suggested by this report, that of overflow. Since it occurs with a much larger probability than do undetected decoding errors, it deserves further examination. In our study of the overflow probability $P_{BF}(N)$ we have found that it is insensitive to buffer size and machine speed, but strongly dependent on signaling rate. This suggests that a sizable decrease in $P_{BF}(N)$ is obtainable only with a decrease in rate. For many applications, large signaling rate is desired. Hence, if $P_{BF}(N)$ could be made to decrease more rapidly with buffer size and machine speed, then the decoder could operate at a higher rate with an equal overflow probability. We are motivated then to consider ways of reducing the size of the "static" computation for each channel noise sequence. As mentioned in Chapter III for Sequential Decoding there exists some high channel noise sequence such that "static" computation is large and growing exponentially with the length of this interval of high channel noise. If the rate of growth of computation with such a channel noise sequence is reduced, then $P_{BF}(N)$ will decrease more rapidly with buffer size and machine speed.

Conceivably, a reduction in the rate of growth of computation with channel noise is possible by modifying the Fano algorithm. If the rate of growth of computation with a modified algorithm remains exponential, then the modified algorithm should be expected to be similar in design and performance to the Fano algorithm. If the rate of growth realized is nonexponential, it is doubtful that the modified algorithm will resemble the Fano algorithm in any way. Exponential growth of computation seems to be characteristic of this algorithm.

If the rate of growth of computation is to be nonexponential, there is some question that the probability of error can be made to decrease with the constraint length of the code S as fast as $2^{-SE(R)}$, as it does for Sequential Decoding algorithms.[7] As a matter of fact, there are a number of decoding procedures for which the computation is bounded by a function which is algebraic in the constraint length or block length S, that is, which grows no faster than S^{β} for some $\beta \geqslant 0$; but at the same time the error probability decreases only as $2^{-S^{1-\epsilon}E(R)}$, where ϵ is some number strictly greater than zero.[3,9,10] There seems to be an important sacrifice in error probability for a reduction in computation. Since a small error probability can be realized with small cost, a trade-off of this type may be desirable. We are prompted to suggest that the obtainable trade-off between computation and error probability is limited by the channel and the signaling rate. If such a trade-off exists, the knowledge of the best balance between computation and error probability would be of great conceptual, and ultimately, practical interest.

Note added in proof: In a recent paper to be published, I. Jacobs and E. Berlekamp through a direct argument have underbounded the probability of a buffer overflow or an undetected error. This bound grows linearly with the number of information digits processed by the decoder and it has as computation exponent that given by the conjecture of this chapter.

Also, H. Yudkin has recently shown that the random code bound of Chapter 4 can be refined so that the lower bound to the computation exponent agrees with the conjectured exponent for rates less than R_{comp}.

APPENDIX
LEMMAS

Lemma 2. (Minkowski's Inequality)

Let $\{w_h\}$, $1 \leqslant h \leqslant H$ be a set of positive random variables. Then,

$$\left[\overline{\left(\sum_{h=1}^{H} w_h\right)^p}\right]^{1/p} \leqslant \sum_{h=1}^{H} (\overline{w_h^p})^{1/p} \quad , \quad p \geqslant 1 \quad .$$

Proof.

Holder's inequality established below, will be used. Write

$$s = w_1 + \ldots + w_H$$

and let $S^p = \overline{s^p}$. Using Holder's Inequality for two variates with $\nu_1 = p$ and $\nu_2 = p/(p-1)$ we have

$$S^p = \sum_{h=1}^{H} \overline{w_h s^{p-1}} \leqslant \sum_{h=1}^{H} (\overline{w_h^p})^{1/p} (\overline{s^p})^{1-1/p} \quad .$$

Then,

$$S^p \leqslant \left[\sum_{h=1}^{H} (\overline{w_h^p})^{1/p}\right] S^{p-1}$$

or

$$S = \left[\overline{\left(\sum_{h=1}^{H} w_h\right)^p}\right]^{1/p} \leqslant \sum_{h=1}^{H} (\overline{w_h^p})^{1/p} \quad . \qquad \text{Q. E. D.}$$

Lemma 7. (Holder's Inequality)

Let $\{w_h\}$, $1 \leqslant h \leqslant H$ be a set of positive random variables and let $\{\nu_h\}$, $1 \leqslant h \leqslant H$ be a set of positive numbers satisfying

$$\sum_{h=1}^{H} \frac{1}{\nu_h} = 1 \quad .$$

Then,

$$\overline{\prod_{h=1}^{H} w_h} \leqslant \prod_{h=1}^{H} \left(\overline{w_h^{\nu_h}}\right)^{1/\nu_h} \quad .$$

Proof.

It suffices to establish that

$$\overline{ab} \leqslant (\overline{a^{\nu}})^{1/\nu} (\overline{b^{\eta}})^{1/\eta} \quad a, b \geqslant 0$$

when $(1/\nu) + (1/\eta) = 1$ since this inequality may be iterated to obtain the inequality of the lemma. Let the joint probability that $a = a_i$ and $b = b_i$ be p_i. Then,

$$\overline{ab} = \sum_i p_i a_i b_i \quad .$$

Let $\Theta(t) = t^{1/\nu} - (1/\nu)\, t$ for $t > 0$. Then,

$$\Theta(t) = \frac{1}{\nu}\,(t^{-1/\eta} - 1) = \begin{cases} > 0 & 0 < t < 1 \\ = 0 & t = 1 \\ < 0 & t > 1 \end{cases} \quad .$$

Therefore, $\Theta(t)$ achieves a maximum at $t = 1$ over the range $t > 0$. Hence,

$$\Theta(t) \leqslant \Theta(1) = \frac{1}{\eta} \quad .$$

Let $t = A/B$ and multiply by B where both A and B are positive to obtain the following

$$A^{1/\nu}\, B^{1/\eta} \leqslant \frac{1}{\nu} A + \frac{1}{\eta} B \quad .$$

Now, choose

$$A = \frac{p_i a_i^{\nu}}{\sum\limits_i p_i a_i^{\nu}} \quad , \quad B = \frac{p_i b_i^{\eta}}{\sum\limits_i p_i b_i^{\eta}} \quad .$$

Replacing A and B by their values and summing on i, we arrive at the desired inequality, namely,

$$\sum_i p_i a_i b_i \leqslant \left(\sum_i p_i a_i^{\nu}\right)^{1/\nu} \left(\sum_i p_i b_i^{\eta}\right)^{1/\eta} \quad . \qquad \text{Q. E. D.}$$

Lemma 8.

As defined below, R_β is a monotone decreasing function of increasing β for $\beta \geqslant 0$.

$$R_\beta = -\frac{1}{\beta} \log_2 \sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/(1+\beta)} \right)^{1+\beta} \quad .$$

Proof.

Let $E(\beta) = \beta R_\beta$. Then,

$$\frac{dR_\beta}{d_\beta} = \frac{d}{d_\beta} \frac{E(\beta)}{\beta} = \frac{\beta E'(\beta) - E(\beta)}{\beta^2} \quad .$$

At $\beta = 0$ the numerator is zero. Its derivative is $\beta E''(\beta)$. We show below that $E''(\beta) \leqslant 0$; hence, the numerator is negative for $\beta \geqslant 0$ as is the derivative of R_β.

To show that $E''(\beta) \leqslant 0$ we shall demonstrate that $E(\beta)$ is a convex upward function.

$$E(\beta) = -\log_2 \sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/(1+\beta)} \right)^{1+\beta} \quad .$$

Holder's Inequality for the two variate case will be used twice. We apply it to the inner sum above with

$$\nu_1 = \frac{1+\beta}{\lambda(1+\beta_1)} \quad , \quad \nu_2 = \frac{1+\beta}{(1-\lambda)(1+\beta_2)}$$

where $0 < \lambda < 1$, $\beta_1, \beta_2 \geqslant 0$ and $\beta = \lambda\beta_1 + (1-\lambda)\,\beta_2$

$$\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/(1+\beta)} \leqslant \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/\lambda(1+\beta_1)} \right)^{\lambda(1+\beta_1)}$$

$$\times \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/[(1-\lambda)(1+\beta_2)]} \right)^{(1-\lambda)(1+\beta_2)}$$

Applying Holder's Inequality to the double sum in the definition of $E(\beta)$ with $\nu_1 = 1/\lambda$, $\nu_2 = 1/(1-\lambda)$ we have

$$\sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/(1+\beta)} \right)^{1+\beta} \leqslant \left[\sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/\lambda(1+\beta_1)} \right)^{1+\beta_1} \right]^{\lambda}$$

$$\times \left[\sum_{j=1}^{J} \left(\sum_{k=1}^{K} p_k p\,[y_j/x_k]^{1/[(1-\lambda)(1+\beta_2)]} \right)^{1+\beta_2} \right]^{1-\lambda} \quad .$$

The inequality is strengthened if the exponents of $p\,[y_j/x_k]$ are replaced by $1/(1+\beta_i)$. Then,

$$E\,[\lambda\beta_1 + (1-\lambda)\,\beta_2] \geqslant \lambda E(\beta_1) + (1-\lambda)\,E(\beta_2)$$

which establishes that $E''(\beta) \leqslant 0$. Q. E. D.

Lemma 9.

Let w be a positive random variable and $0 < \nu \leqslant \eta$. Then,

$$(\overline{w^\nu})^{1/\nu} \leqslant (\overline{w^\eta})^{1/\eta} \quad .$$

Proof.

Let $w = w_i$ with probability p_i, then,

$$(\overline{w^\nu})^{1/\nu} = \left(\sum_i p_i w_i^\nu \right)^{1/\nu} \quad .$$

We have

$$\frac{d}{d\nu}\,(\overline{w^{\nu}})^{1/\nu} = -\frac{1}{\nu^2}\,(\overline{w^{\nu}})^{1/\nu}\ln\left(\sum_i p_i w_i^{\nu}\right) + \frac{1}{\nu}\,(\overline{w^{\nu}})^{1/\nu - 1}\left(\sum_i p_i w_i^{\nu}\ln w_i\right)$$

$$= \frac{1}{\nu^2}\,(\overline{w^{\nu}})^{1/\nu}\left(\sum_i p_i Q_i \ln Q_i\right)$$

where

$$Q_i = \frac{w_i^{\nu}}{\sum_i p_i w_i^{\nu}} \geqslant 0$$

and

$$\sum_i p_i Q_i = 1 \quad .$$

Using the standard inequality $\ln x \geqslant 1 - (1/x)$, the derivative is lower bounded by

$$\frac{d}{d\nu}\,(\overline{w^{\nu}})^{1/\nu} \geqslant \frac{1}{\nu^2}\,(\overline{w^{\nu}})^{1/\nu}\left[\sum_i p_i Q_i\left(1 - \frac{1}{Q_i}\right)\right] = 0 \quad . \qquad \text{Q. E. D.}$$

Lemma 10.

The function $\sigma_o R - \mu_p(\sigma_o)$ where $\mu_p(\sigma_o)$ is given by

$$\mu_p(\sigma_o) = \frac{1}{1+p}\log_2 \sum_{j=1}^{J} f(y_j)\left[\sum_{k=1}^{K} p_k\left(\frac{p\,[y_j/x_k]}{f(y_j)}\right)^{1+\sigma_o}\right]^{1+p}$$

is positive for $\sigma' \leqslant \sigma_o \leqslant 0$ where σ' is such that $\mu_p(\sigma')/\sigma' = R$, and $\mu_p(\sigma_o)/\sigma_o$ is monotone increasing in σ_o.

Proof.

For $\sigma_o R - \mu_p(\sigma_o)$ to be positive we must have $R \leqslant \mu_p(\sigma_o)/\sigma_o$ since $\sigma_o \leqslant 0$. If $\mu_p(\sigma_o)/\sigma_o$ is monotone increasing in σ_o, the desired result is established. We shall now show that such is true. The derivative

$$\frac{d}{d\sigma_o}\,\frac{\mu_p(\sigma_o)}{\sigma_o} = \frac{\sigma_o\mu_p'(\sigma_o) - \mu_p(\sigma_o)}{\sigma_o^2}$$

is positive if the numerator is positive. Since the numerator is zero for $\sigma_o = 0$ it suffices to show that its derivative, $\sigma_o\mu_p''(\sigma_o)$, is negative for $\sigma_o \leqslant 0$ or $\mu_p''(\sigma_o) \geqslant 0$.

Let $a_{jk} = p\,[y_j/x_k]/f(y_j)$. Then,

$$\frac{\mu_p'(\sigma_o)}{\log_2 e} = \frac{\sum_{j=1}^{J} f(y_j)\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\right)^{p}\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\ln a_{jk}\right)}{2^{(1+p)\mu_p(\sigma_o)}}$$

also,

$$\frac{\mu_p''(\sigma_o)}{\log_2 e} = \frac{p \sum_{j=1}^{J} f(y_j)\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\right)^{p-1}\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o} \ln a_{jk}\right)^2}{2^{(1+p)\mu_p(\sigma_o)}}$$

$$+ \frac{\sum_{j=1}^{J} f(y_j)\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\right)^{p}\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o} (\ln a_{jk})^2\right)}{2^{(1+p)\mu_p(\sigma_o)}}$$

$$-(1+p)\left[\frac{\sum_{j=1}^{J} f(y_j)\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\right)^{p}\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o} \ln a_{jk}\right)}{2^{(1+p)\mu_p(\sigma_o)}}\right]^2$$

If we let

$$q_{jk} = \frac{f(y_j)\left(\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}\right)^{p} p_k a_{jk}^{1+\sigma_o}}{2^{(1+p)\mu_p(\sigma_o)}}$$

$$h_j = \sum_{k=1}^{K} q_{jk}$$

both of which are "tilted" probabilities and let

$$\varphi_j = \frac{\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o} \ln a_{jk}}{\sum_{k=1}^{K} p_k a_{jk}^{1+\sigma_o}}$$

then, we have

$$\frac{\mu_p''(\sigma_o)}{\log_2 e} = p\left[\sum_{j=1}^{J} h_j(\varphi_j)^2 - \left(\sum_{j=1}^{J} h_j\varphi_j\right)^2\right]$$

$$+ \left[\sum_{j=1}^{J}\sum_{k=1}^{K} q_{jk}(\ln a_{jk})^2 - \left(\sum_{j=1}^{J}\sum_{k=1}^{K} q_{jk} \ln a_{jk}\right)^2\right]$$

which is positive because both terms are variances. Therefore, $\mu_p(\sigma_o)/\sigma_o$ is monotone increasing in increasing σ_o. Q. E. D.

ACKNOWLEDGMENTS

I wish to sincerely thank Professor I.M. Jacobs for his continuing interest in this problem and for his many helpful comments and criticisms. I also express my gratitude to Professors J.M. Wozencraft, R.G. Gallager and C.E. Shannon for the many helpful discussions I have had with them during the course of the research and the writing of this document. My association with these four scholars has been the most rewarding aspect of my graduate study.

I thank the Research Laboratory of Electronics, M.I.T., and Lincoln Laboratory for financial support of this work.

REFERENCES

1. C.E. Shannon and W. Weaver, Mathematical Theory of Communication (University of Illinois Press, Urbana, Illinois, 1949).
2. J.L. Massey, "Threshold Decoding," Technical Report 410, Research Laboratory of Electronics, M.I.T. (5 April 1963).
3. R.G. Gallager, Low Density Parity Check Codes (M.I.T. Press, Cambridge, Mass., 1963).
4. W.W. Peterson, Error-Correcting Codes (M.I.T. Press, Cambridge, Mass., and Wiley, New York, 1961).
5. P. Elias, "Error-Free Coding," Trans. IRE, PGIT IT-4, 29 (1954).
6. I.M. Jacobs, "Optimum Error Detection Codes for Noiseless Decision Feedback," Trans. IRE, PGIT IT-8 (1962).
7. J.M. Wozencraft and B. Reiffen, Sequential Decoding (M.I.T. Press, Cambridge, Mass., and Wiley, New York, 1961).
8. R.M. Fano, "A Heuristic Discussion of Probabilistic Decoding," Trans. IEEE, PTGIT IT-9, 64 (1963).
9. J. Ziv, "A New Efficient Coding and Decoding Scheme for Memoryless Channels" (to be published).
10. D. Forney, "Concatenated Codes," ScD Thesis, Department of Electrical Engineering, M.I.T. (June 1965).
11. B. Reiffen, "Sequential Encoding and Decoding for the Discrete Memoryless Channel," Technical Report 374, Research Laboratory of Electronics, M.I.T. (August 1960); Technical Report 231, Lincoln Laboratory, M.I.T. (12 August 1960), DDC 247612, H-146.

12. I.L. Lebow, "A Qualitative Description of Sequential Decoding," Group Report 62G-4, Lincoln Laboratory, M.I.T. (12 July 1963), DDC 413949, H-529.

13. G. Blustein and K. L. Jordan, Jr., "An Investigation of the Fano Sequential Decoding Algorithm by Computer Simulation," Group Report 62G-5, Lincoln Laboratory, M. I. T. (12 July 1963), DDC 412632, H-525.

14. R. M. Fano, unpublished class notes presented during Spring Semester, 1963, at Massachusetts Institute of Technology.

15. I. G. Stiglitz, "Sequential Decoding with Feedback," PhD Thesis, Department of Electrical Engineering, M. I. T. (August 1963).

16. H.L. Yudkin, "Channel State Testing in Information Decoding," PhD Thesis, Department of Electrical Engineering, M.I.T. (September 1964).

17. R.G. Gallager, "Lower Bounds on the Tails of Probability Distributions," Quarterly Progress Report No.75, Research Laboratory of Electronics, M.I.T. (15 January 1965).

18. C.E. Shannon, unpublished seminar notes presented in 1956.

19. W. Feller, An Introduction to Probability Theory and its Applications (Wiley, New York, 1957), Chapter 2.

20. R.M. Fano, Transmission of Information (M.I.T. Press, Cambridge, Mass., and Wiley, New York, 1961), p. 136.

21. J.M. Wozencraft, "List Decoding," Quarterly Progress Report No. 48, Research Laboratory of Electronics, M.I.T. (15 January 1958).

22. P. Elias, "List Decoding For Noisy Channels," Technical Report No. 335, Research Laboratory of Electronics, M. I. T. (20 September 1957).

23. R. G. Gallager, unpublished notes.

24. R. M. Fano, op. cit., Chapter 9.

25. I. Stiglitz, op. cit., p. 32.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Lincoln Laboratory, M.I.T.

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
None

3. REPORT TITLE
The Computation Problem with Sequential Decoding

4. OESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report

5. AUTHOR(S) *(Last name, first name, initial)*
Savage, John E.

6. REPORT OATE
16 February 1965

7a. TOTAL NO. OF PAGES
84

7b. NO. OF REFS
25

8a. CONTRACT OR GRANT NO.
AF 19(628)-500

b. PROJECT NO.

c. None

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Technical Report 371

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
ESD-TDR-65-52; Research Laboratory of Electronics Technical Report 439

10. AVAILABILITY/LIMITATION NOTICES
None

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Air Force Systems Command, USAF

13. ABSTRACT

Fano Sequential Decoding is a technique for communicating at a high information rate and with a high reliability over a large class of channels. However, equipment cost and variation in the time required to decode successive transmitted digits limit its use. This work is concerned with the latter limitation.

Others have shown that the average processing time per decoded digit is small if the information rate of the source is less than a rate R_{comp}. This report studies the probability distribution of the processing time random variable and applies the results to the buffer overflow probability. It is shown that the overflow probability is relatively insensitive to the buffer storage capacity and to the computational speed of the decoder but quite sensitive to information rate. Halving the source rate more than squares the overflow probability. These sensitivities are found to be basic to Sequential Decoding and arise because the computation per decoded digit is large during an interval of high channel noise and grows exponentially with the length of such an interval.

A conjecture is presented concerning the exact behavior of the overflow probability with information rate. This conjecture agrees well with the (limited) experimental evidence available.

14. KEY WOROS

sequences
decoding
probability
distribution
data processing
storage
random variable